Tools for Networkers v4.2

HyungKwang Choi (2020.10)

Table of Contents

1. Introduction

While working as a TAC, I've experienced some common needs. It's about small programs which helpful & convenient at a practical usage perspective. Hope this to help others who are not good at operating Juniper router product or troubleshooting and python programing.

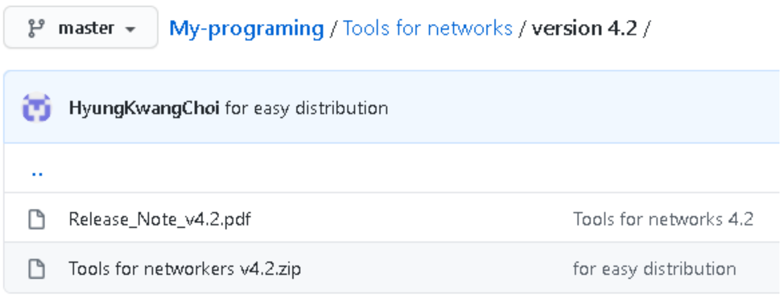
1.1 Building Environment

** This Tools has been coded and working in Window10. MAC or other OS may not work.

a). Make a folder and install Python3.8.2 in it, download 'Tools for networkers v4.2.zip' file and extract it under the folder you created.

```
- After you make a folder such as 'Python38', install python in it.
- Then set PATH env properly.
- Download: https://github.com/HyungKwangChoi/My-programing
- Place the file under your folder, and must extract 'Tools for networkers
v4.2.zip'.
```

- From github



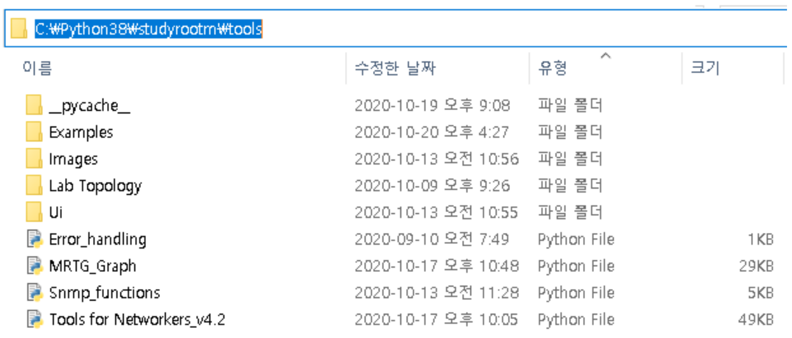
master / My-programing / Tools for networks / version 4.2 /

HyungKwangChoi for easy distribution

..

Release_Note_v4.2.pdf	Tools for networks 4.2
Tools for networkers v4.2.zip	for easy distribution

- From your python folder, all files must be seen like below after extracting '**Tools for networkers v4.2.zip**'



C:₩Python38₩studyrootm₩tools

이름	수정한 날짜	유형	크기
__pycache__	2020-10-19 오후 9:08	파일 폴더	
Examples	2020-10-20 오후 4:27	파일 폴더	
Images	2020-10-13 오전 10:56	파일 폴더	
Lab Topology	2020-10-09 오후 9:26	파일 폴더	
Ui	2020-10-13 오전 10:55	파일 폴더	
Error_handling	2020-09-10 오전 7:49	Python File	1KB
MRTG_Graph	2020-10-17 오후 10:48	Python File	29KB
Snmp_functions	2020-10-13 오전 11:28	Python File	5KB
Tools for Networkers_v4.2	2020-10-17 오후 10:05	Python File	49KB

b). Install Python & libraries.

```
- python 3.8.2
- pyqt5: GUI library
- pyqt5-tool (QT designer): used for GUI.
- scapy: used for 'Packet builder'
- pygtgraph: used for 'MRTG' real-time plotting
- pysnmp: used for 'MRTG' asyncio snmp query
- matplotlib: used for 'MRTG' restoring figures
- pandas: used for 'MRTG', help to restore figures by manipulating csv file
dataframe.
- numpy: used for pandas example test. (This is an option. It doesn't
affect to this tool running)
```

```
PS C:\Temp> python --version
Python 3.8.2

C:\Users\test>cd C:\Python38\studyrootm\tools

C:\Python38\studyrootm\tools>pip3 install pyqt5
C:\Python38\studyrootm\tools>pip3 install pyqt5-tools
C:\Python38\studyrootm\tools>pip3 install pyqtgraph
C:\Python38\studyrootm\tools>pip3 install scapy
C:\Python38\studyrootm\tools>pip3 install pysnmp
C:\Python38\studyrootm\tools>pip3 install matplotlib
C:\Python38\studyrootm\tools>pip3 install pandas
C:\Python38\studyrootm\tools>pip3 install numpy

C:\Python38\studyrootm\tools>pip3 list
Package          Version
----------------- --------------------
matplotlib        3.2.1
pandas            1.0.5
PyQt5             5.15.0
PyQt5-sip         12.8.1
pyqt5-tools       5.15.0.1.7.1
pyqtgraph         0.11.0.dev0+g932b975
pysnmp            4.4.12
scapy             2.4.3
numpy             1.18.2
```

c). Run it to see if it's running or not.

```
C:\Python38\studyrootm\tools>dir

2020-10-20  오후 04:27    <DIR>          Examples
2020-10-13  오전 10:56    <DIR>          Images
2020-10-09  오후 09:26    <DIR>          Lab Topology
2020-10-13  오전 10:55    <DIR>          Ui
2020-09-10  오전 07:49               372 Error_handling.py
```

```
2020-10-17  오후 10:48          29,126 MRTG_Graph.py
2020-10-13  오전 11:28           4,899 Snmp_functions.py
2020-10-17  오후 10:05          50,159 Tools for Networkers_v4.2.py

C:\Python38\studyrootm\tools>python "Tools for Networkers_v4.2.py"
```

1.2 Revision History

Revision	Date	Author	Comments
1.1	Dec, 2019	hkchoi	Modified some Telnet bug at ‘Repetitive Tasks’
2.1	Jan, 2020	hkchoi	New feature added naming ‘Packet Builder’
2.2	Feb, 2020	hkchoi	Auto scroll down feature added on the right display screen of the Tool.
3.2	Feb, 2020	hkchoi	New feature added naming ‘Easy Lab Replication’
4.2	Sep, 2020	hkchoi	New feature added naming ‘MRTG’

1.3 Download

Revision	Download
1.1	https://github.com/HyungKwangChoi/My-programing/tree/master/Tools%20for%20networks/version%201.0
2.1	https://github.com/HyungKwangChoi/My-programing/tree/master/Tools%20for%20networks/version%202.1
2.2	https://github.com/HyungKwangChoi/My-programing/tree/master/Tools%20for%20networks/version%202.1
3.2	https://github.com/HyungKwangChoi/My-programing/tree/master/Tools%20for%20networks/version%203.2
4.2	https://github.com/HyungKwangChoi/My-programing/tree/master/Tools%20for%20networks/version%204.2

1.4 Referenced

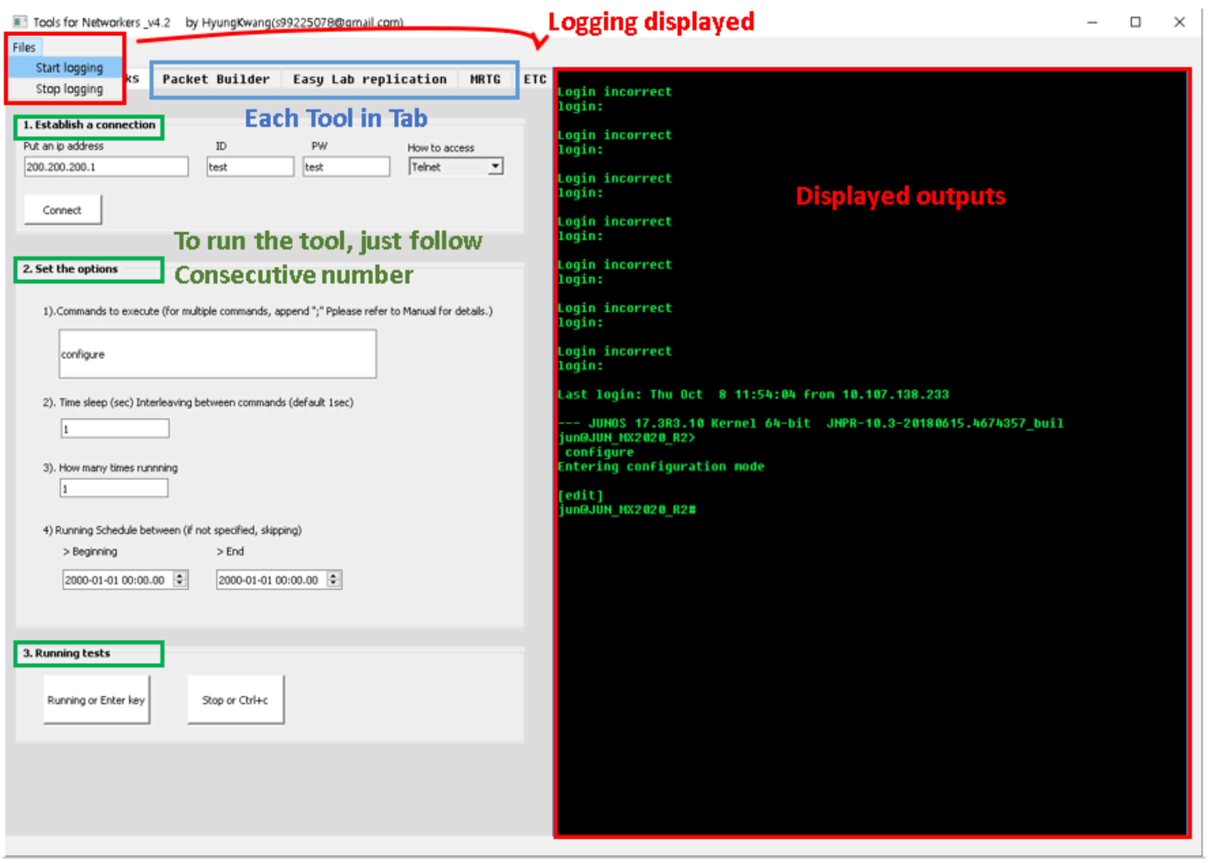
Bookmark	Title / Author / Link	Rev
ctypes	https://stackoverflow.com/questions/2963263/how-can-i-create-a-simple-message-box-in-python	
pandas dataframe merge	https://rfriend.tistory.com/m/258	
python shared memory	https://docs.python.org/3/library/multiprocessing.shared_memory.html	
pyqtgraph bug	https://github.com/pyqtgraph/pyqtgraph/issues/1077	
git window download	https://git-scm.com/download/win	

2. Tools

2.1 Intro (Design view and common function)

There 3 major parts of this tool from display perspective. And all tools running method has similar patterns (just following consecutive number.)

1. All system outputs are displayed on the right black screen.
2. Tabs which showing each tool.
3. Basically, running each tool is very simple
   Just follow the consecutive number to run a tool.



1. Establish a connection

Put an ip address: 200.200.200.1
ID: test
PW: test
How to access: Telnet

2. Set the options

1).Commands to execute (for multiple commands, append ";" Pplease refer to Manual for details.)
configure

2). Time sleep (sec) Interleaving between commands (default 1sec)
1

3). How many times running
1

4) Running Schedule between (if not specified, skipping)
> Beginning: 2000-01-01 00:00.00
> End: 2000-01-01 00:00.00

3. Running tests

```
Login incorrect
login:

Login incorrect
login:

Login incorrect
login:

Login incorrect
login:

Login incorrect
login:

Login incorrect
login:

Login incorrect
login:

Last login: Thu Oct  8 11:54:04 from 10.107.138.233

--- JUNOS 17.3R3.10 Kernel 64-bit  JNPR-10.3-20180615.4674357_buil
jun@JUN_MX2020_R2>
 configure
Entering configuration mode

[edit]
jun@JUN_MX2020_R2#
```

2.2 ‘Repetitive Tasks’

2.2.1 Overview

This tool is to help someone who feels hard to create a Shell script or other scripts to run commands repetitively. It’s implemented via “Telnet” access. So, this tool could be used for various propose if you are able to access/log in via telnet.

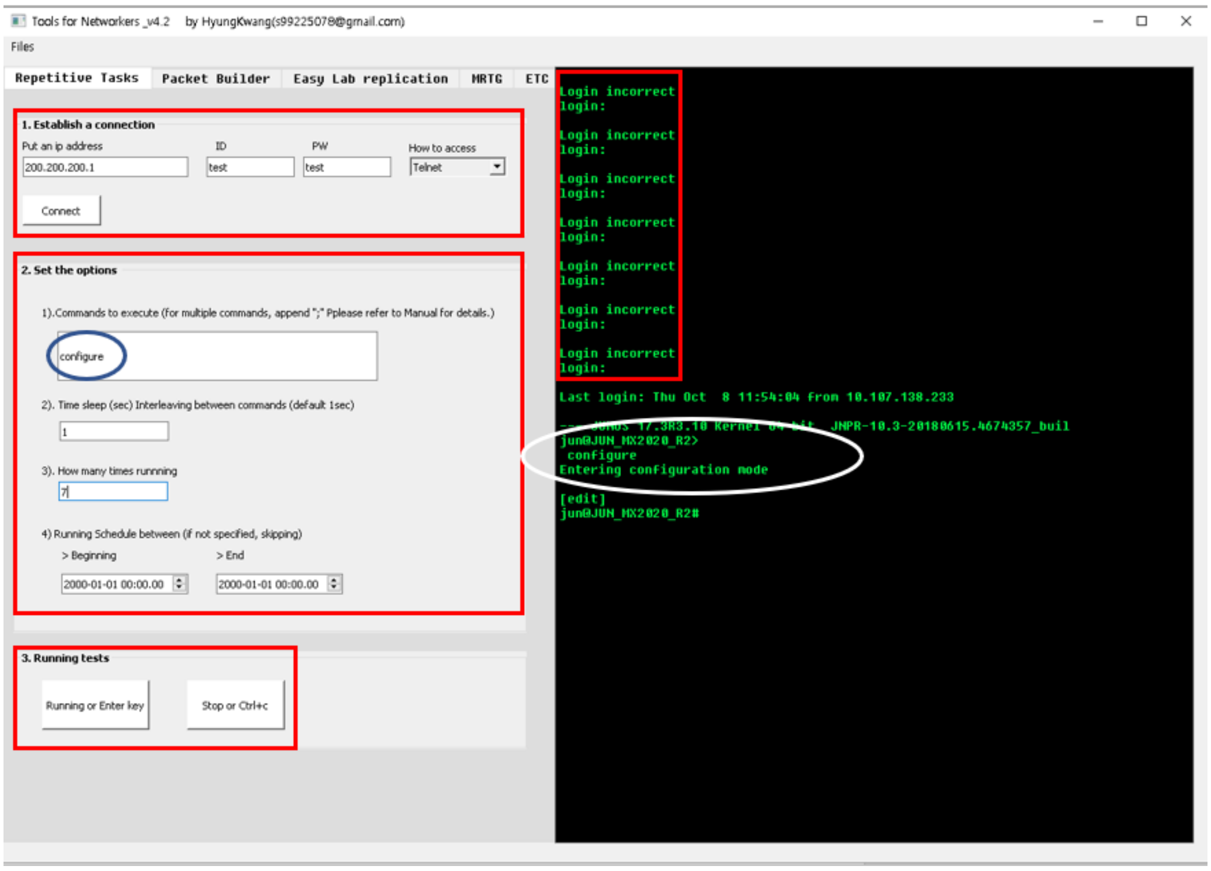
2.2.1.1 Library used in this tool (telnetlib)

This is Python basic lib.You don’t need to install it.

```
import telnetlib
```

2.2.2 How to run

It’s simple.
Just follow the consecutive number step by step.



1. Establish a connection

Put an ip address	ID	PW	How to access
200.200.200.1	test	test	Telnet

2. Set the options

1).Commands to execute (for multiple commands, append ";" Pplease refer to Manual for details.): configure

2). Time sleep (sec) Interleaving between commands (default 1sec): 1

3). How many times running: 7

4) Running Schedule between (if not specified, skipping)
> Beginning: 2000-01-01 00:00.00
> End: 2000-01-01 00:00.00

3. Running tests

```
Login incorrect
login:

Login incorrect
login:

Login incorrect
login:

Login incorrect
login:

Login incorrect
login:

Login incorrect
login:

Login incorrect
login:
Last login: Thu Oct  8 11:54:04 from 10.107.138.233

--- JUNOS 17.3R3.10 Kernel 64-bit  JNPR-10.3-20180615.4674357_buil
jun@JUN_MX2020_R2>
 configure
Entering configuration mode

[edit]
jun@JUN_MX2020_R2#
```

1. Put ip & id & pw
- Telnet connection Timeout = 3
- In my source-code, login/Password prompts are “login”, “Password” coded.
  So, if your device doesn’t return the Char as “login”, “Password”, please modify my source-code.

2. If connection succeed,
- Look at *1). commands to execute (for multiple command, append “;”)*
If you want to run several commands, you have to add “;” among commands

ex) set interface ge-0/0/0 mtu 9000; commit; delete interface ge-0/0/0 mtu 9000; run show interface terse | match ge-0/0/0

- If possible, try to append '**no-more**' after commands. To avoid logs are stuck in buffer just like the same as Telnet session.
  ex) show version | no-more, show log messages | no-more

- If needed, put the time-sleep which is time interval between commands executed.

- If needed, put the number at "how many times to run". This is a kind of loop.

3. "Running tests"

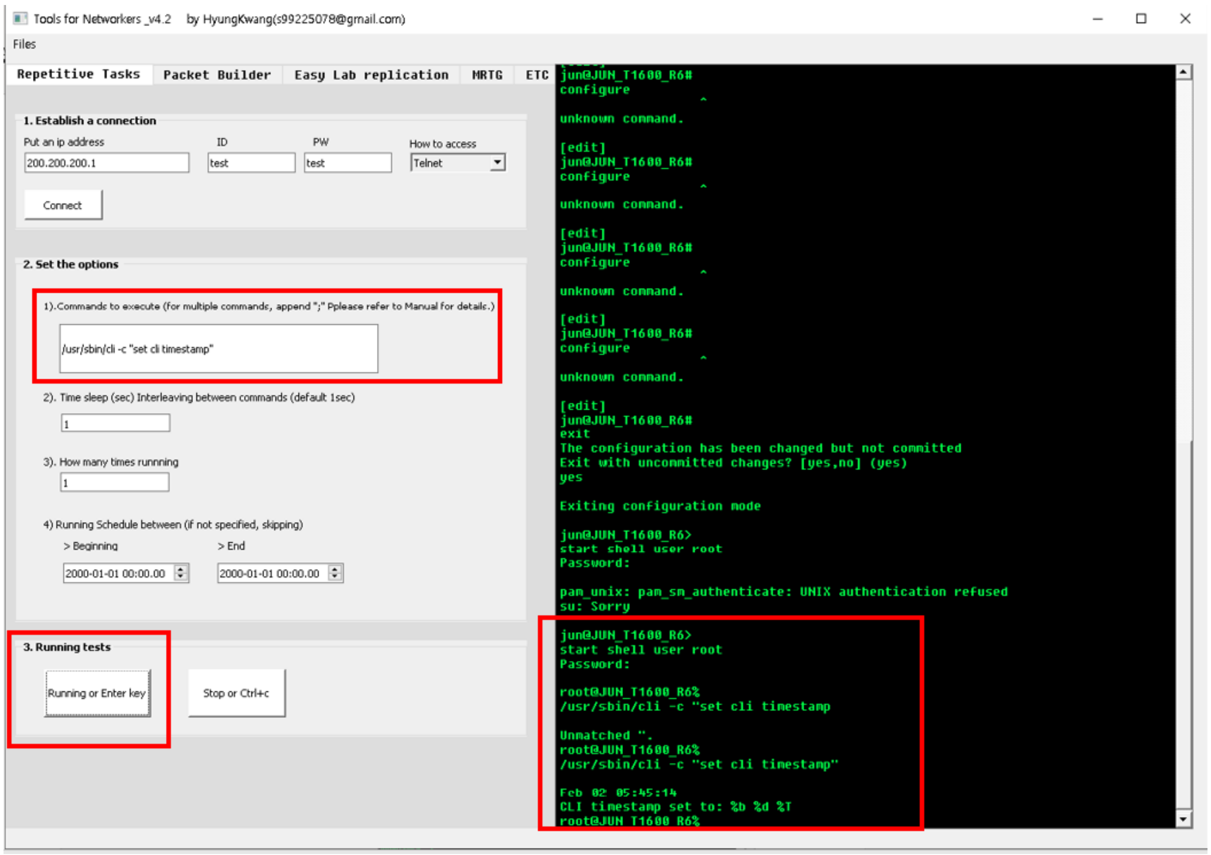
2.2.3 How to login a system and execute shell or cli commands

** The device here used for an example is from Juniper product (running JUNOS)

- As this 'Repetitive Tasks' works on Telnet session, you can do the same thing as what you do in Telnet. (**Accessing to System, and executing shell command or whatever**)

  Ex)
  1. put "start shell user root;juniper" and "Running or Enter key". if you login to the system successfully
  2. put "/usr/sbin/cli -c "set cli timestamp" and "Running or Enter key".

- Additionally, if the device is Juniper router or Linux box ****and**** If you want to run python script directly in Juniper product(FreeBSD) or Linux box, your python commands have to run on Shell such as) csh, bash…or else. for details, please refer to session "3.1.2 Running shell commands"



Tools for Networkers _v4.2 by HyungKwang(s99225078@gmail.com)
Files
Repetitive Tasks
Packet Builder
Easy Lab replication
MRTG
ETC
1. Establish a connection
Put an ip address: 200.200.200.1
ID: test
PW: test
How to access: Telnet
Connect
2. Set the options
1).Commands to execute (for multiple commands, append ";" Ppleace refer to Manual for details.)
/usr/sbin/cli -c "set cli timestamp"
2). Time sleep (sec) Interleaving between commands (default 1sec)
1
3). How many times running
1
4) Running Schedule between (if not specified, skipping)
> Beginning: 2000-01-01 00:00.00
> End: 2000-01-01 00:00.00
3. Running tests
Running or Enter key
Stop or Ctrl+c

```
jun@JUN_T1600_R6#
configure
                    ^
unknown command.

[edit]
jun@JUN_T1600_R6#
configure
                    ^
unknown command.

[edit]
jun@JUN_T1600_R6#
configure
                    ^
unknown command.

[edit]
jun@JUN_T1600_R6#
configure
                    ^
unknown command.

[edit]
jun@JUN_T1600_R6#
exit
The configuration has been changed but not committed
Exit with uncommitted changes? [yes,no] (yes)
yes

Exiting configuration mode

jun@JUN_T1600_R6>
start shell user root
Password:

pam_unix: pam_sm_authenticate: UNIX authentication refused
su: Sorry

jun@JUN_T1600_R6>
start shell user root
Password:

root@JUN_T1600_R6%
/usr/sbin/cli -c "set cli timestamp

Unmatched ".
root@JUN_T1600_R6%
/usr/sbin/cli -c "set cli timestamp"

Feb 02 05:45:14
CLI timestamp set to: %b %d %T
root@JUN_T1600_R6%
```

2.3 ‘Packet builder’

2.3.1 Overview

Working as a TAC, I needed a tool to manipulate packets and send it in various ways.
There already are many freeware tools for packet builder. But there are some limitations which are already compiled. So, you can’t do further based on you needs. So..hope you can do change codes and use it for your own purpose.

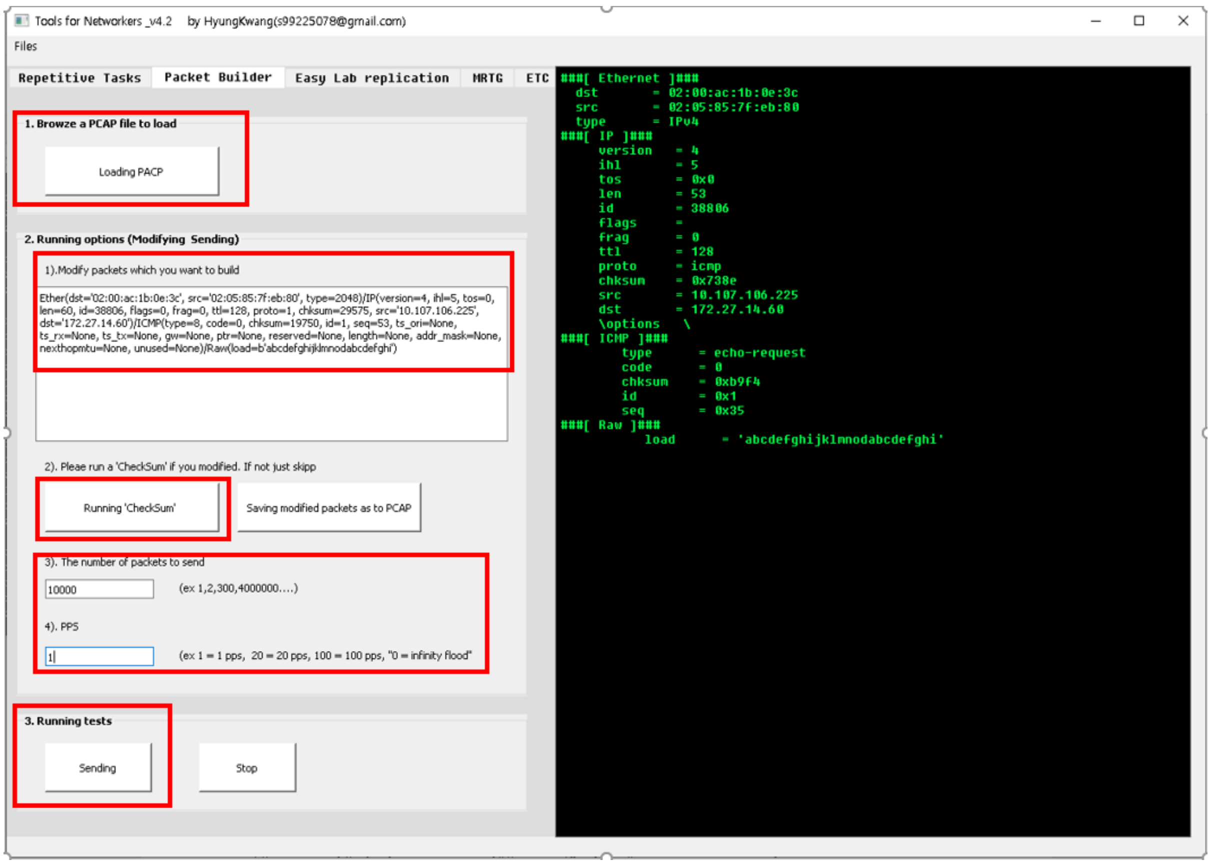
2.3.1.1 Library used (scapy)

This tool basically built with SCAPY lib. SCAPY provides a lot of features to manipulate packets and send or else. So, by studying SCAPY lib further, you can evolve/enhance this tool with your own purpose.
To use it, you must install scapy.

```
C:\Python38\studyrootm\tools>pip3 install scapy

C:\Python38\studyrootm\tools>pip3 list
Package          Version
----------------- --------------------
scapy            2.4.3

#### From the code  ####
from scapy.all import *
```

2.3.2 How to run

It’s simple and intuitive.
Just follow the consecutive number step by step.


Tools for Networkers _v4.2 by HyungKwang(s99225078@gmail.com)
Files
Repetitive Tasks
Packet Builder
Easy Lab replication
MRTG
ETC
1. Browze a PCAP file to load
Loading PACP
2. Running options (Modifying Sending)
1).Modify packets which you want to build
2). Pleae run a 'CheckSum' if you modified. If not just skipp
Running 'CheckSum'
Saving modified packets as to PCAP
3). The number of packets to send
(ex 1,2,300,4000000....)
4). PPS
(ex 1 = 1 pps, 20 = 20 pps, 100 = 100 pps, "0 = infinity flood"
3. Running tests
Sending
Stop

```
Ether(dst='02:00:ac:1b:0e:3c', src='02:05:85:7f:eb:80', type=2048)/IP(version=4, ihl=5, tos=0,
len=60, id=38806, flags=0, frag=0, ttl=128, proto=1, chksum=29575, src='10.107.106.225',
dst='172.27.14.60')/ICMP(type=8, code=0, chksum=19750, id=1, seq=53, ts_ori=None,
ts_rx=None, ts_tx=None, gw=None, ptr=None, reserved=None, length=None, addr_mask=None,
nexthopmtu=None, unused=None)/Raw(load=b'abcdefghijklmnodabcdefghi')
```

```
10000
```

```
1
```

```
###[ Ethernet ]###
  dst       = 02:00:ac:1b:0e:3c
  src       = 02:05:85:7f:eb:80
  type      = IPv4
###[ IP ]###
     version   = 4
     ihl       = 5
     tos       = 0x0
     len       = 53
     id        = 38806
     flags     =
     frag      = 0
     ttl       = 128
     proto     = icmp
     chksum    = 0x738e
     src       = 10.107.106.225
     dst       = 172.27.14.60
     \options   \
###[ ICMP ]###
        type      = echo-request
        code      = 0
        chksum    = 0xb9f4
        id        = 0x1
        seq       = 0x35
###[ Raw ]###
           load      = 'abcdefghijklmnodabcdefghi'
```

1. load PCAP files

2. Once you loaded PCAP files, you can see packet lines at white text box.
- Then modify each packet value or else whatever you want to manipulate.
- Once you are done modifying the packet text lines, then run "Running Checksum".

*** while running "checksum" if you make a mistake or else, you are going to see the error messages on the right black screen. For further details, please refer to below "2.3.3 Errors"

- Then put the number of packets to send and pps.

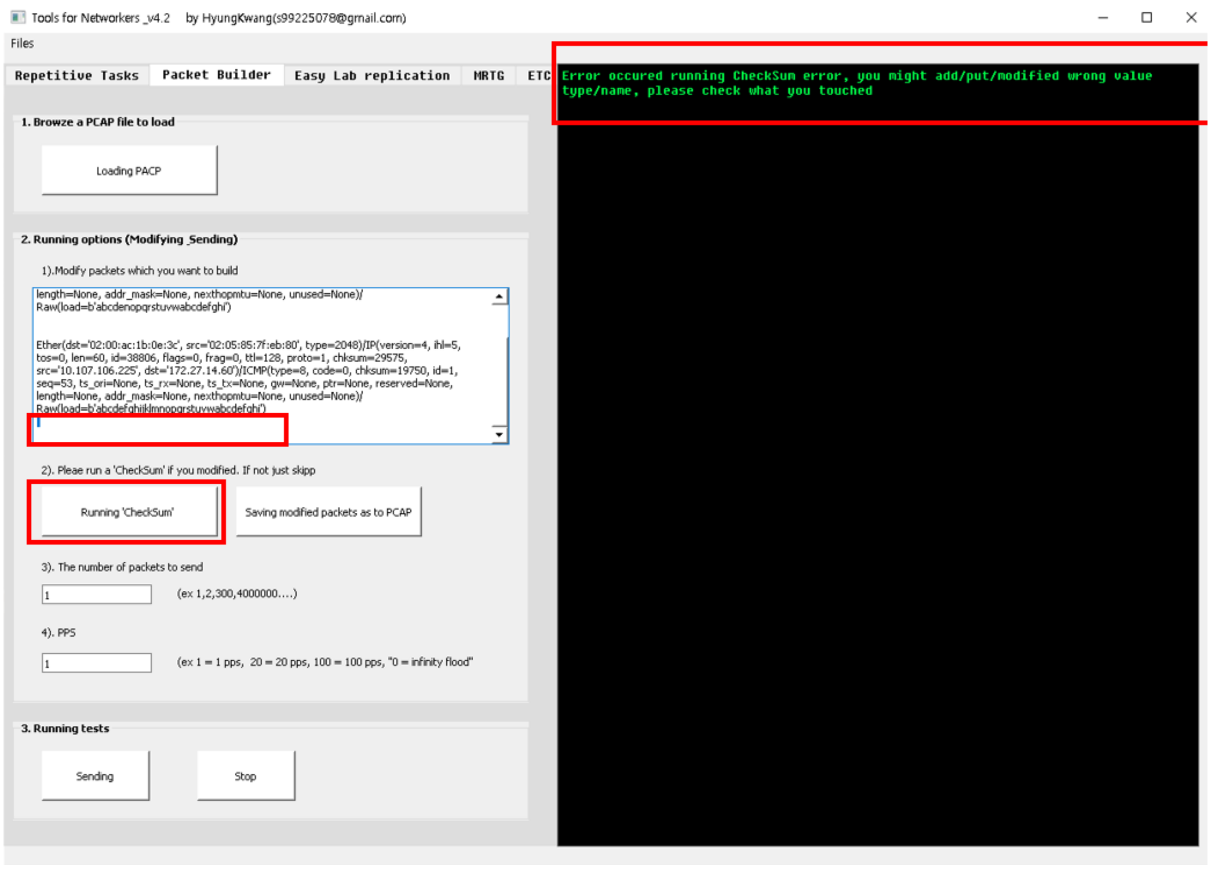
3. "Running tests"

2.3.3 Errors

1. "Error occurred running 'CheckSum' error, you might add/put/modified wrong value type/name, please check what you touched"

After you modified, when you executed "Running Chceksum", you are going to see the error messages displayed. That means while you are modifying packets, you made a mistake what ever.

➢ **How to resolve :**
Please remove "empty space" or "\n" "\r" at the last line of packets. That causes the problem.

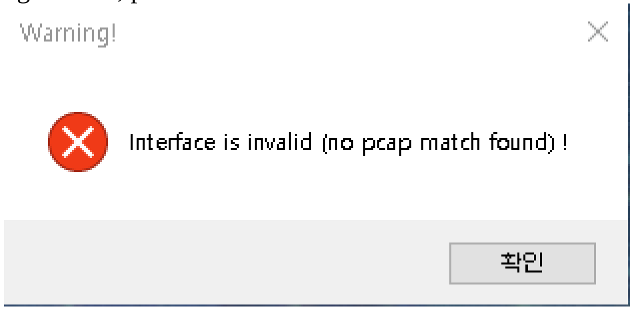

Tools for Networkers _v4.2   by HyungKwang(s99225078@gmail.com)
Files
Repetitive Tasks
Packet Builder
Easy Lab replication
MRTG
ETC
1. Browze a PCAP file to load
Loading PACP
2. Running options (Modifying _Sending)
1).Modify packets which you want to build
2). Pleae run a 'CheckSum' if you modified. If not just skipp
Running 'CheckSum'
Saving modified packets as to PCAP
3). The number of packets to send
(ex 1,2,300,4000000....)
4). PPS
(ex 1 = 1 pps, 20 = 20 pps, 100 = 100 pps, "0 = infinity flood"
3. Running tests
Sending
Stop

```
length=None, addr_mask=None, nexthopmtu=None, unused=None)/
Raw(load=b'abcdenopqrstuvwabcdefghi')


Ether(dst='02:00:ac:1b:0e:3c', src='02:05:85:7f:eb:80', type=2048)/IP(version=4, ihl=5,
tos=0, len=60, id=38806, flags=0, frag=0, ttl=128, proto=1, chksum=29575,
src='10.107.106.225', dst='172.27.14.60')/ICMP(type=8, code=0, chksum=19750, id=1,
seq=53, ts_ori=None, ts_rx=None, ts_tx=None, gw=None, ptr=None, reserved=None,
length=None, addr_mask=None, nexthopmtu=None, unused=None)/
Raw(load=b'abcdefghijklmnopqrstuvwabcdefghi')
```

```
Error occured running CheckSum error, you might add/put/modified wrong value
type/name, please check what you touched
```

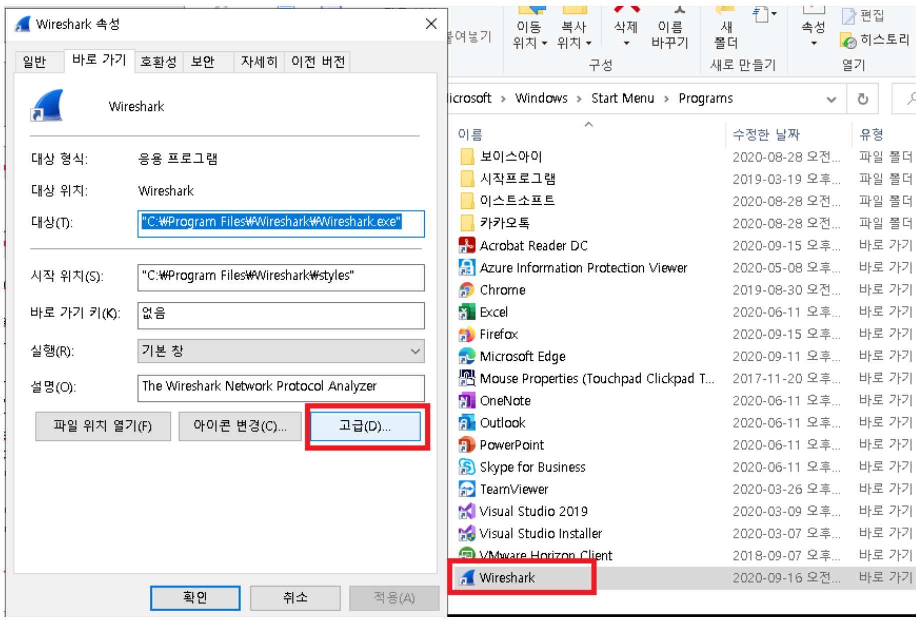
2. "Interface is invalid (no pcap match found) ! "

That means wireshark has to be run in Administrator.
So, before running the tool, please run wireshark in Administor first.



Interface is invalid (no pcap match found) !

- **How to resolve**
  Please check "Running the Wireshark in Admin permission"



Wireshark 속성

일반 | 바로 가기 | 호환성 | 보안 | 자세히 | 이전 버전

Wireshark

대상 형식: 응용 프로그램

대상 위치: Wireshark

대상(T): "C:₩Program Files₩Wireshark₩Wireshark.exe"

시작 위치(S): "C:₩Program Files₩Wireshark₩styles"

바로 가기 키(K): 없음

실행(R): 기본 창

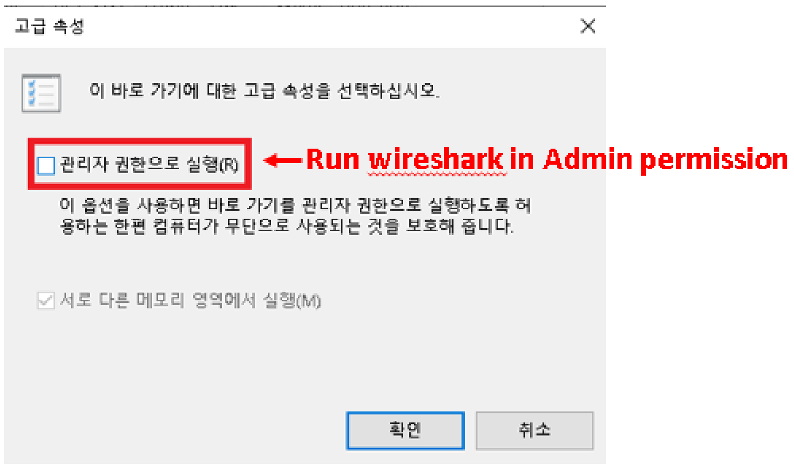
설명(O): The Wireshark Network Protocol Analyzer

파일 위치 열기(F) | 아이콘 변경(C)... | 고급(D)...

확인 | 취소 | 적용(A)

icrosoft › Windows › Start Menu › Programs

이름	수정한 날짜	유형
보이스아이	2020-08-28 오전...	파일 폴더
시작프로그램	2019-03-19 오후...	파일 폴더
이스트소프트	2020-08-28 오전...	파일 폴더
카카오톡	2020-08-28 오전...	파일 폴더
Acrobat Reader DC	2020-09-15 오후...	바로 가기
Azure Information Protection Viewer	2020-05-08 오후...	바로 가기
Chrome	2019-08-30 오전...	바로 가기
Excel	2020-06-11 오후...	바로 가기
Firefox	2020-09-15 오후...	바로 가기
Microsoft Edge	2020-09-11 오후...	바로 가기
Mouse Properties (Touchpad Clickpad T...	2017-11-20 오후...	바로 가기
OneNote	2020-06-11 오후...	바로 가기
Outlook	2020-06-11 오후...	바로 가기
PowerPoint	2020-06-11 오후...	바로 가기
Skype for Business	2020-06-11 오후...	바로 가기
TeamViewer	2020-03-26 오후...	바로 가기
Visual Studio 2019	2020-03-09 오후...	바로 가기
Visual Studio Installer	2020-03-07 오후...	바로 가기
VMware Horizon Client	2018-09-07 오후...	바로 가기
Wireshark	2020-09-16 오전...	바로 가기



고급 속성

이 바로 가기에 대한 고급 속성을 선택하십시오.

☐ 관리자 권한으로 실행(R)

Run wireshark in Admin permission

이 옵션을 사용하면 바로 가기를 관리자 권한으로 실행하도록 허용하는 한편 컴퓨터가 무단으로 사용되는 것을 보호해 줍니다.

☑ 서로 다른 메모리 영역에서 실행(M)

확인 | 취소

2.4 'Easy Lab Replication'

2.4.1 Overview

The purpose of this tool is to provide various lab configuration.

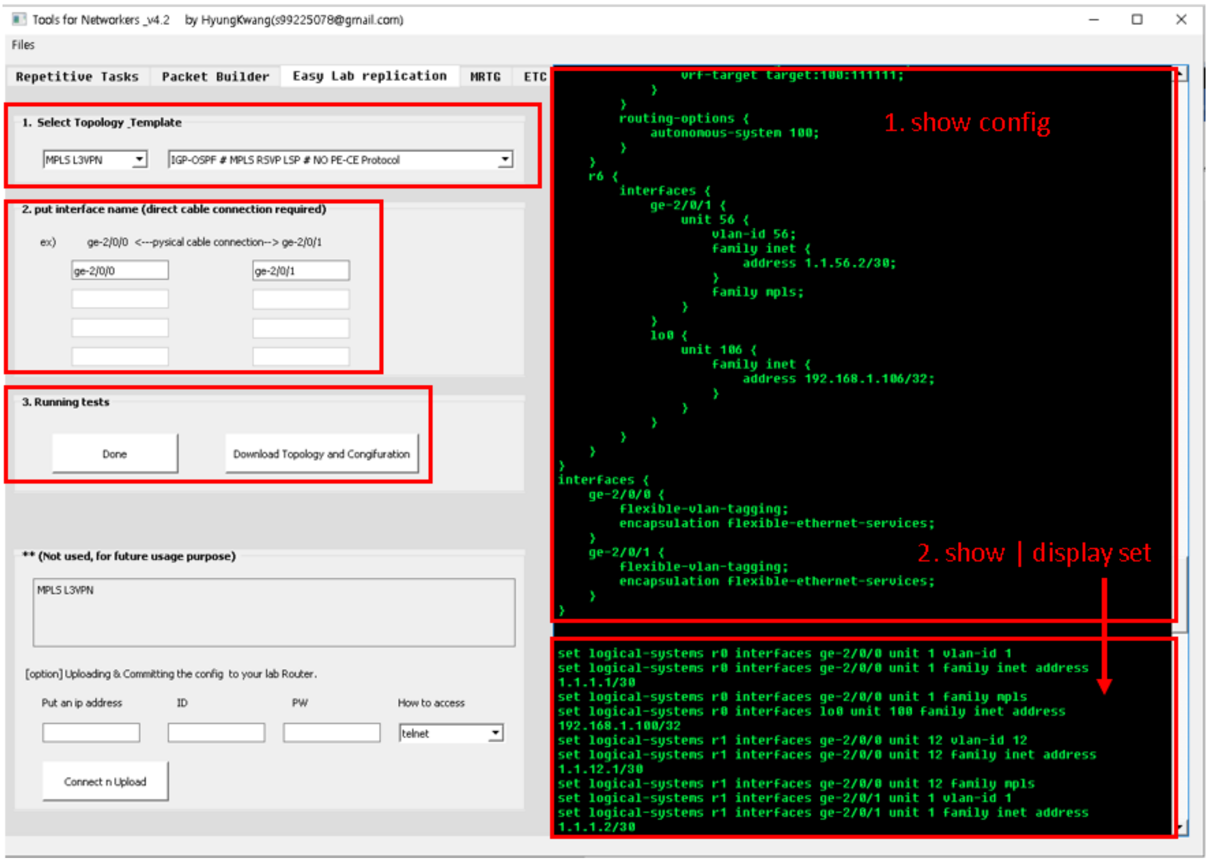
- This tool usage is dedicated to Juniper Router product. Because, each lab scenario configuration consists of logical routers, which feature only supported in Juniper router product.

- Aims :
  1). To help easy lab replication with only one Router box, in case with juniper router.
  2). With 30 x lab scenario, you can study various routing protocols & features with easy.

- Total 30 Lab scenario (for details, please refer to "Note" below)
  MPLS L3VPN : 14
  MPLS L2VPN : 2
  MPLS VPLS : 2
  Multicast : 5
  6PE : 1
  6VPE : 1
  INTRA-AS VPN : 3
  Various scenario: 2

2.4.2 How to run

It's simple and intuitive.
Just follow the consecutive number step by step.



Tools for Networkers _v4.2 by HyungKwang(s99225078@gmail.com)
Files
Repetitive Tasks
Packet Builder
Easy Lab replication
MRTG
ETC
1. Select Topology_Template
MPLS L3VPN
IGP-OSPF # MPLS RSVP LSP # NO PE-CE Protocol
2. put interface name (direct cable connection required)
ex) ge-2/0/0 <---pysical cable connection--> ge-2/0/1
ge-2/0/0
ge-2/0/1
3. Running tests
Done
Download Topology and Congifuration
** (Not used, for future usage purpose)
MPLS L3VPN
[option] Uploading & Committing the config to your lab Router.
Put an ip address
ID
PW
How to access
telnet
Connect n Upload
1. show config
2. show | display set

```
                vrf-target target:100:111111;
            }
        }
        routing-options {
            autonomous-system 100;
        }
    }
    r6 {
        interfaces {
            ge-2/0/1 {
                unit 56 {
                    vlan-id 56;
                    family inet {
                        address 1.1.56.2/30;
                    }
                    family mpls;
                }
            }
            lo0 {
                unit 106 {
                    family inet {
                        address 192.168.1.106/32;
                    }
                }
            }
        }
    }
}
interfaces {
    ge-2/0/0 {
        flexible-vlan-tagging;
        encapsulation flexible-ethernet-services;
    }
    ge-2/0/1 {
        flexible-vlan-tagging;
        encapsulation flexible-ethernet-services;
    }
}
```

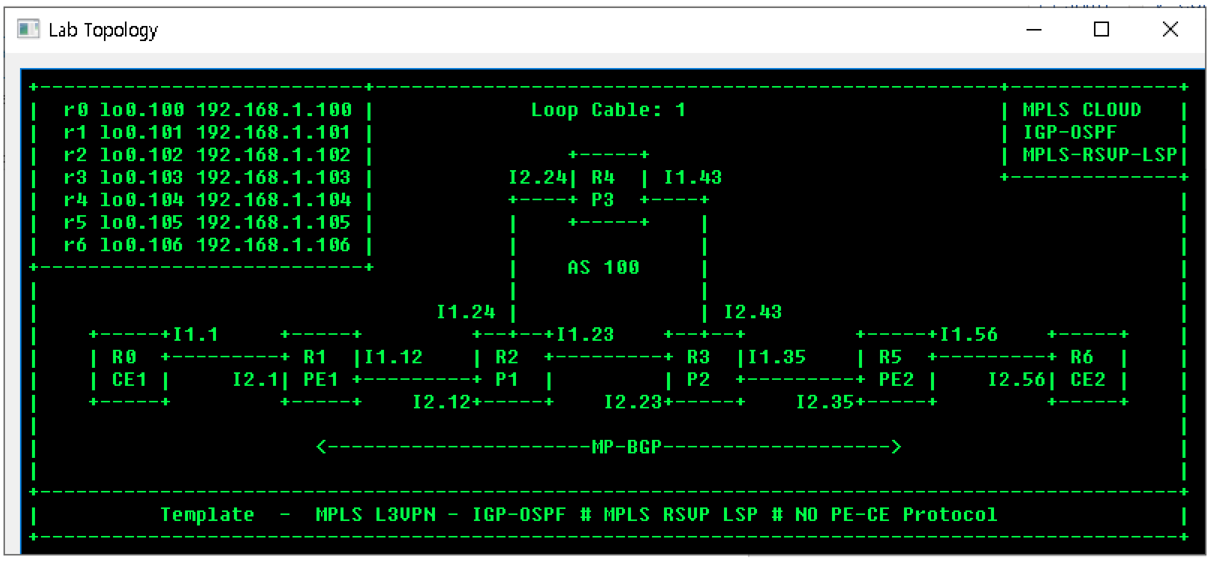
```
set logical-systems r0 interfaces ge-2/0/0 unit 1 vlan-id 1
set logical-systems r0 interfaces ge-2/0/0 unit 1 family inet address
1.1.1.1/30
set logical-systems r0 interfaces ge-2/0/0 unit 1 family mpls
set logical-systems r0 interfaces lo0 unit 100 family inet address
192.168.1.100/32
set logical-systems r1 interfaces ge-2/0/0 unit 12 vlan-id 12
set logical-systems r1 interfaces ge-2/0/0 unit 12 family inet address
1.1.12.1/30
set logical-systems r1 interfaces ge-2/0/0 unit 12 family mpls
set logical-systems r1 interfaces ge-2/0/1 unit 1 vlan-id 1
set logical-systems r1 interfaces ge-2/0/1 unit 1 family inet address
1.1.1.2/30
```

1. First select lab topo and template.
*** When you selected "lab topo and template", simple lab template will pop-up without specific interface naming. (see below figure)

2. Based on "lab topo and template" you selected, put the physical interface naming you will have loop connection.
- un-editable gray input box turns into editable.
- the number physical connection required differs from topo type.

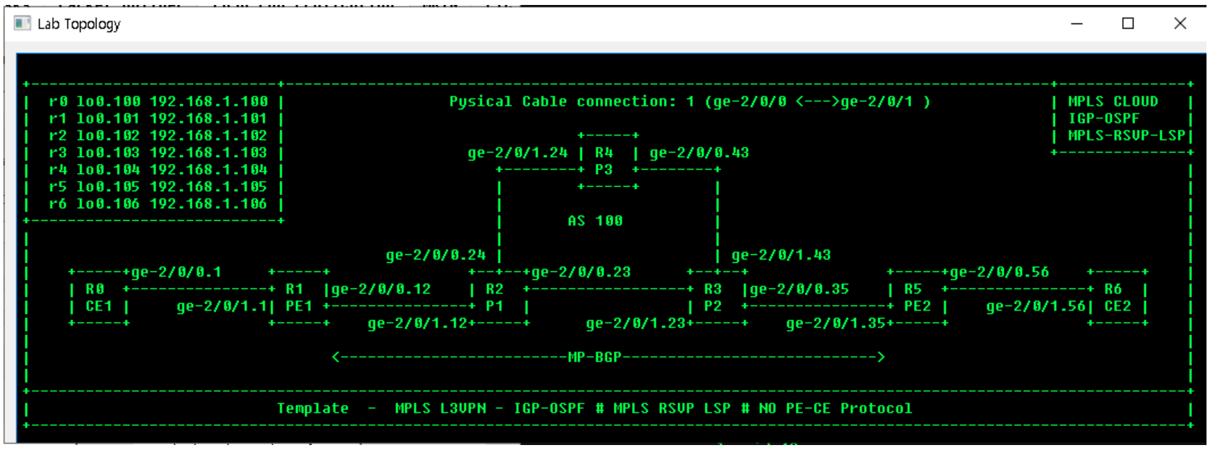
3. After you put the interface name correctly, window pop-up will be refreshed displaying it's interface naming on the template (see below figure). Then it will give you configurations on the right display box.
- There are 2 types of configuration displayed :
  The first one : "show configuration"
  The second : "show | display set"

> Simple lab template without specific interface naming.



```
+---------------------------+------------------------------------------------+---------------+
|  r0 lo0.100 192.168.1.100 |             Loop Cable: 1                      | MPLS CLOUD    |
|  r1 lo0.101 192.168.1.101 |                                                | IGP-OSPF      |
|  r2 lo0.102 192.168.1.102 |                +-----+                         | MPLS-RSVP-LSP|
|  r3 lo0.103 192.168.1.103 |          I2.24| R4   | I1.43                   +---------------+
|  r4 lo0.104 192.168.1.104 |          +----+ P3   +----+                                    |
|  r5 lo0.105 192.168.1.105 |          |     +-----+    |                                    |
|  r6 lo0.106 192.168.1.106 |          |                |                                    |
+---------------------------+          |     AS 100     |                                    |
|                                      |                |                                    |
|                               I1.24 |                | I2.43                              |
|    +-----+I1.1      +-----+     +--+--+I1.23    +--+--+         +-----+I1.56    +-----+    |
|    | R0  +---------+ R1  |I1.12   | R2  +---------+ R3  |I1.35   | R5  +---------+ R6  |    |
|    | CE1 |    I2.1| PE1 +---------+ P1  |         | P2  +---------+ PE2 |    I2.56| CE2 |    |
|    +-----+        +-----+   I2.12+-----+     I2.23+-----+    I2.35+-----+         +-----+    |
|                                                                                             |
|                       <----------------------MP-BGP------------------->                     |
|                                                                                             |
+---------------------------------------------------------------------------------------------+
|          Template  -  MPLS L3VPN - IGP-OSPF # MPLS RSVP LSP # NO PE-CE Protocol             |
+---------------------------------------------------------------------------------------------+
```

> After you put the interface name correctly, window pop-up will be refreshed



```
+---------------------------+----------------------------------------------------------------------+----------------+
|  r0 lo0.100 192.168.1.100 |                 Pysical Cable connection: 1 (ge-2/0/0 <--->ge-2/0/1 )               | MPLS CLOUD     |
|  r1 lo0.101 192.168.1.101 |                                                                      | IGP-OSPF       |
|  r2 lo0.102 192.168.1.102 |                         +-----+                                      | MPLS-RSVP-LSP|
|  r3 lo0.103 192.168.1.103 |             ge-2/0/1.24 | R4  | ge-2/0/0.43                          +----------------+
|  r4 lo0.104 192.168.1.104 |              +---------+ P3  +---------+                                              |
|  r5 lo0.105 192.168.1.105 |              |          +-----+         |                                              |
|  r6 lo0.106 192.168.1.106 |              |                          |                                              |
+---------------------------+              |         AS 100           |                                              |
|                                          |                          |                                              |
|                              ge-2/0/0.24 |                          | ge-2/0/1.43                                  |
|    +-----+ge-2/0/0.1      +-----+      +--+--+ge-2/0/0.23      +--+--+           +-----+ge-2/0/0.56    +-----+    |
|    | R0  +----------------+ R1  |ge-2/0/0.12   | R2  +----------------+ R3  |ge-2/0/0.35   | R5  +----------------+ R6  |    |
|    | CE1 |     ge-2/0/1.1| PE1 +----------------+ P1  |                | P2  +----------------+ PE2 |     ge-2/0/1.56| CE2 |    |
|    +-----+               +-----+    ge-2/0/1.12+-----+      ge-2/0/1.23+-----+    ge-2/0/1.35+-----+                +-----+    |
|                                                                                                                    |
|                         <-------------------------MP-BGP------------------------->                                 |
|                                                                                                                    |
+--------------------------------------------------------------------------------------------------------------------+
|                    Template  -  MPLS L3VPN - IGP-OSPF # MPLS RSVP LSP # NO PE-CE Protocol                          |
+--------------------------------------------------------------------------------------------------------------------+
```

2.4.3 Total 30 lab scenario in details

Main Feature	Subset
MPLS L3VPN	IGP-OSPF # MPLS RSVP LSP # NO PE-CE Protocol
	IGP-OSPF # MPLS RSVP LSP # PE-CE Protocol OSPF # NO RR
	IGP-OSPF # MPLS RSVP LSP # PE-CE Protocol BGP # NO RR
	IGP-OSPF # MPLS RSVP LSP # PE-CE Protocol OSPF # SINGLE RR
	IGP-OSPF # MPLS RSVP LSP # PE-CE Protocol BGP # SINGLE RR
	IGP-OSPF # MPLS RSVP LSP # PE-CE Protocol OSPF # DUAL RR
	IGP-OSPF # MPLS RSVP LSP # PE-CE Protocol BGP # DUAL RR
	IGP-OSPF # MPLS LDP # NO PE-CE Protocol
	IGP-OSPF # MPLS LDP # PE-CE Protocol OSPF # NO RR
	IGP-OSPF # MPLS LDP # PE-CE Protocol BGP # NO RR
	IGP-OSPF # MPLS LDP # PE-CE Protocol OSPF # SINGLE RR
	IGP-OSPF # MPLS LDP # PE-CE Protocol BGP # SINGLE RR
	IGP-OSPF # MPLS LDP # PE-CE Protocol OSPF # DUAL RR
	IGP-OSPF # MPLS LDP # PE-CE Protocol BGP # DUAL RR
MPLS L2VPN	MPLS L2VPN - IGP-OSPF # MPLS RSVP LSP # SINGLE RR [3 loops Required]
	MPLS L2VPN - IGP-OSPF # MPLS LDP # SINGLE RR [3 loops Required]
MPLS VPLS	VPLS [3 CE] - IGP OSPF # MPLS RSVP LSP # SINGLE RR [4 loops Required]
	VPLS [3 CE] - IGP OSPF # MPLS LDP # SINGLE RR [4 loops Required]
Multicast	PLAIN MULTICAST # PIM - SPARSE MODE # STATIC RP CONFIGURED # ASM
	PLAIN MULTICAST # PIM - DENCE MODE
	PLAIN MULTICAST # PIM - SPARSE MODE # NO RP CONFIGURED # SSM
	NG-MVPN # MBGP Multicast VPN with PIM SSM as PE-CE Protocol
	NG-MVPN # MBGP Multicast VPN with PIM ASM as PE-CE Protocol # RPT-SPT
6PE	6PE - IGP-OSPF # MPLS RSVP LSP # PE-CE Protocol BGP # SINGLE RR
6VPE	6VPE - IGP-OSPF # MPLS RSVP LSP # PE-CE Protocol BGP # SINGLE RR
INTER-AS VPN	INTER-AS L3VPN OPTION-A IGP-OSPF # MPLS LDP # PE-CE Protocol EBGP
	INTER-AS L3VPN OPTION-B IGP-OSPF # MPLS LDP # PE-CE Protocol EBGP
	INTER-AS L3VPN OPTION-C IGP-OSPF # MPLS LDP # PE-CE Protocol EBGP
Various scenario	IGP-OSPF -- MULTI-AREA - STUB - NSSA -TSA -ALL P2MP LINKS [1 loop]
	IGP-OSPF -- MULTI-AREA - STUB - NSSA -TSA -ALL P2P LINKS [1 loop]

2.5 ‘MRTG’

2.5.1 Overview

Working in TAC, for various troubleshooting purpose, sometimes I used to MRTG graph. But usually 3rd vendor or free MRTG graph doesn’t meet my needs. They are already all compiled and we couldn’t adjust or modify some options or else. So, I decided to make it for my own usage purpose.
For this tool, I used various library. The reason why “to use my code for other usage ”.

- **This tool has 2 main features.**
  1. **“Real-time plotting”**
  2. **“restoring figures”**

2.5.1.1 Library used (pysnmp, pandas, matplotlib, pyqtgraph)

1. pysnmp lib : used for snmp function, and you must install it.
Asyncio feature used to run a function asynchronous. Because most of all functions run in synchronization. That means “when a function runs, it blocks I/O and hold CPU resource until the function returns something(complete or error or else)”. So, for the multiprocessing environment, to run something asynchronous, we have to run it in asynchronization.

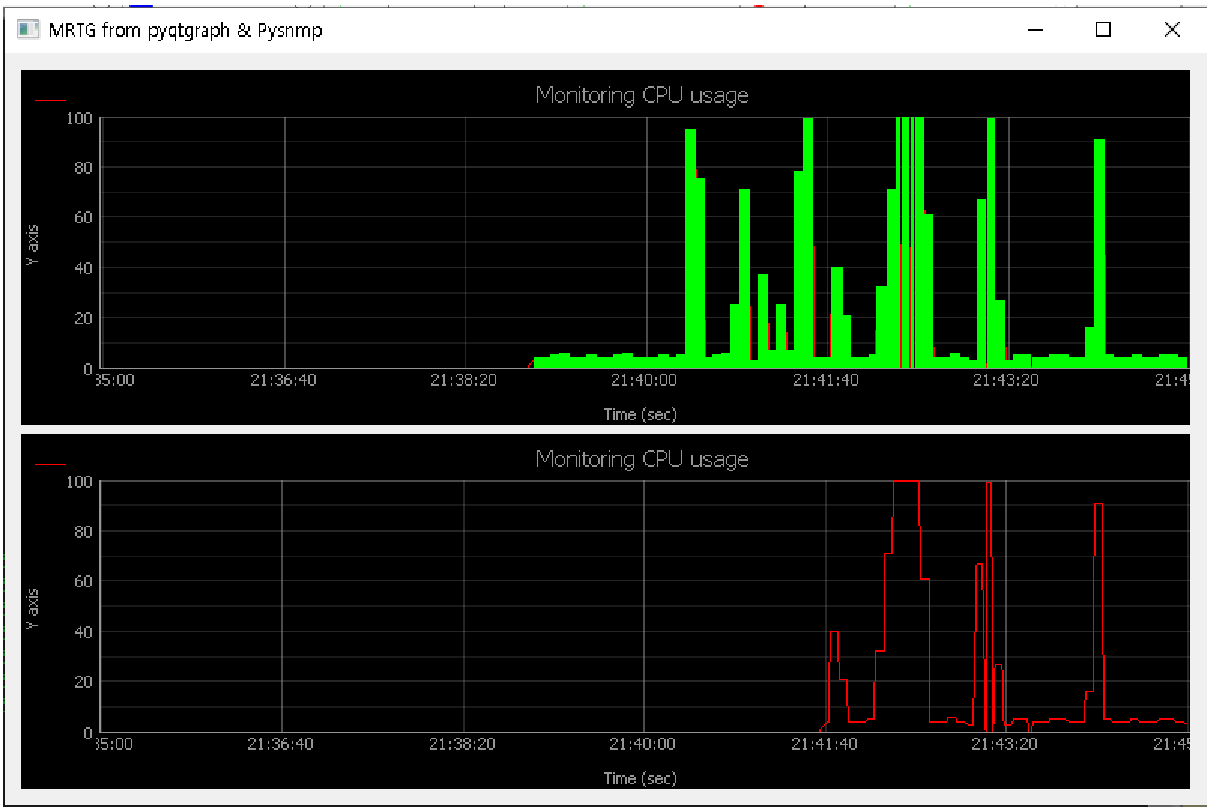
2. pandas : used to manipulate csv data. It’s very powerful tool for data manipulation.
And you must install it.

3. matplotlib : used for restoring figures. Matplotlib is a representative tool drawing a figure.
This function used to restore figures in my tool. Because matplotlib provide a lot of figure options.
And you must install it.

4. pygtgraph : This lib used for real-time plotting. The reason why I used this lib instead of matplotlib, this is very light and suitable for real-time plotting. At the first time I used matplotlib for realtime plotting. But matplotlib is heavy, and showed some delay realtime plotting in multi-threading environment. So, I used pyqtgraph lib.
And you must install it.

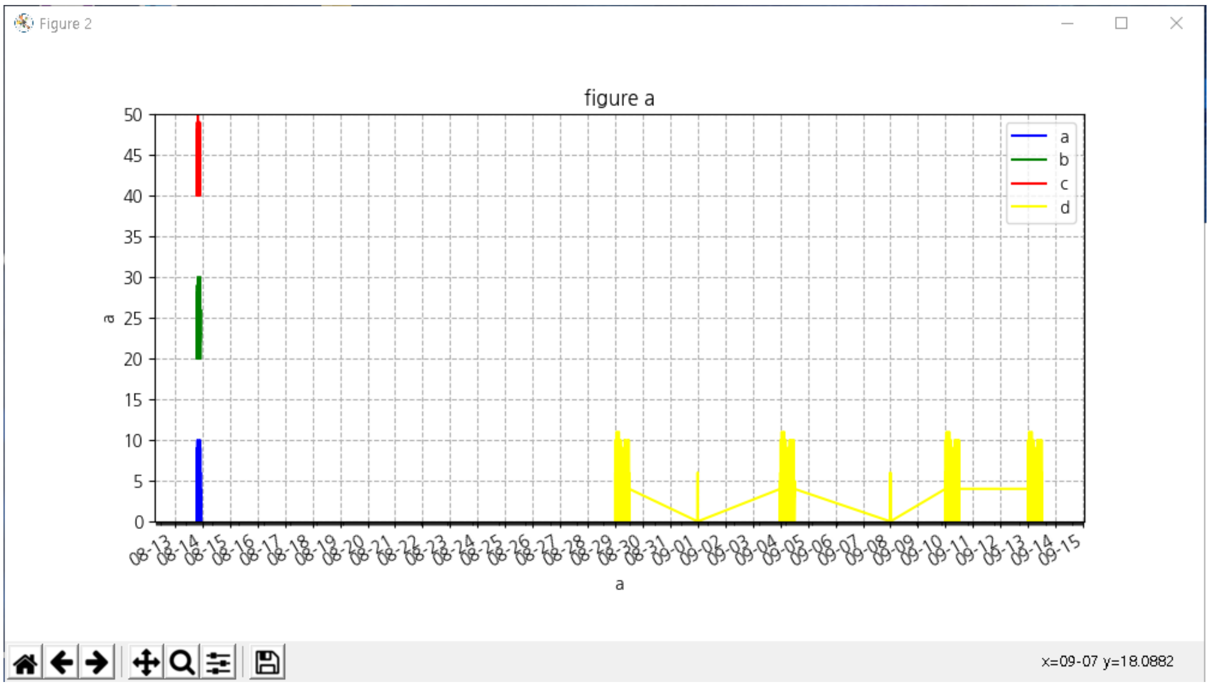
2.5.1.2 2major features (“real-time plotting”, “restoring figures”)

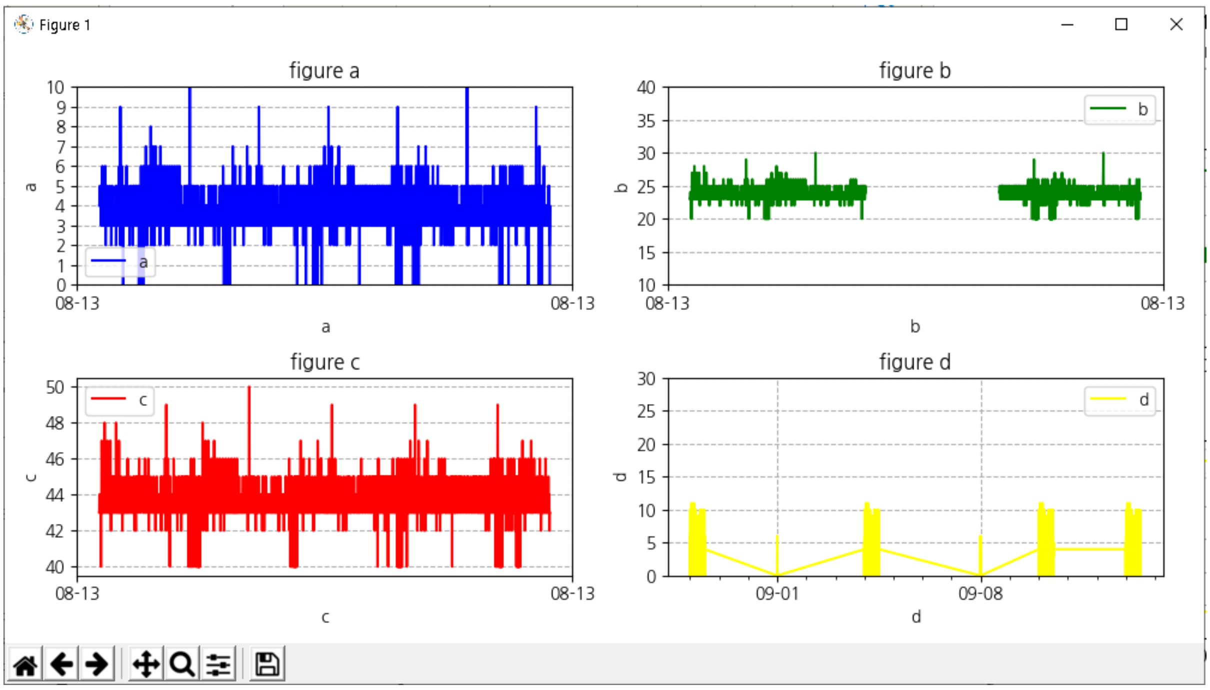
1. Real-time plotting per sec.
It’s implemented with pyqtgraph lib and pysnmp lib.
I am planning to enhance this feature further with various options in the future. (adjusting time interval, adding an average plotting, and displaying real-time OID return to right black screen)



2. Restoring figures.

When you load csv files, it draws a figure in a one shot or several subplots.
It's implemented with matplotlib and pandas.





2.5.2 How to run

It's simple and intuitive.
Just follow the consecutive number step by step.

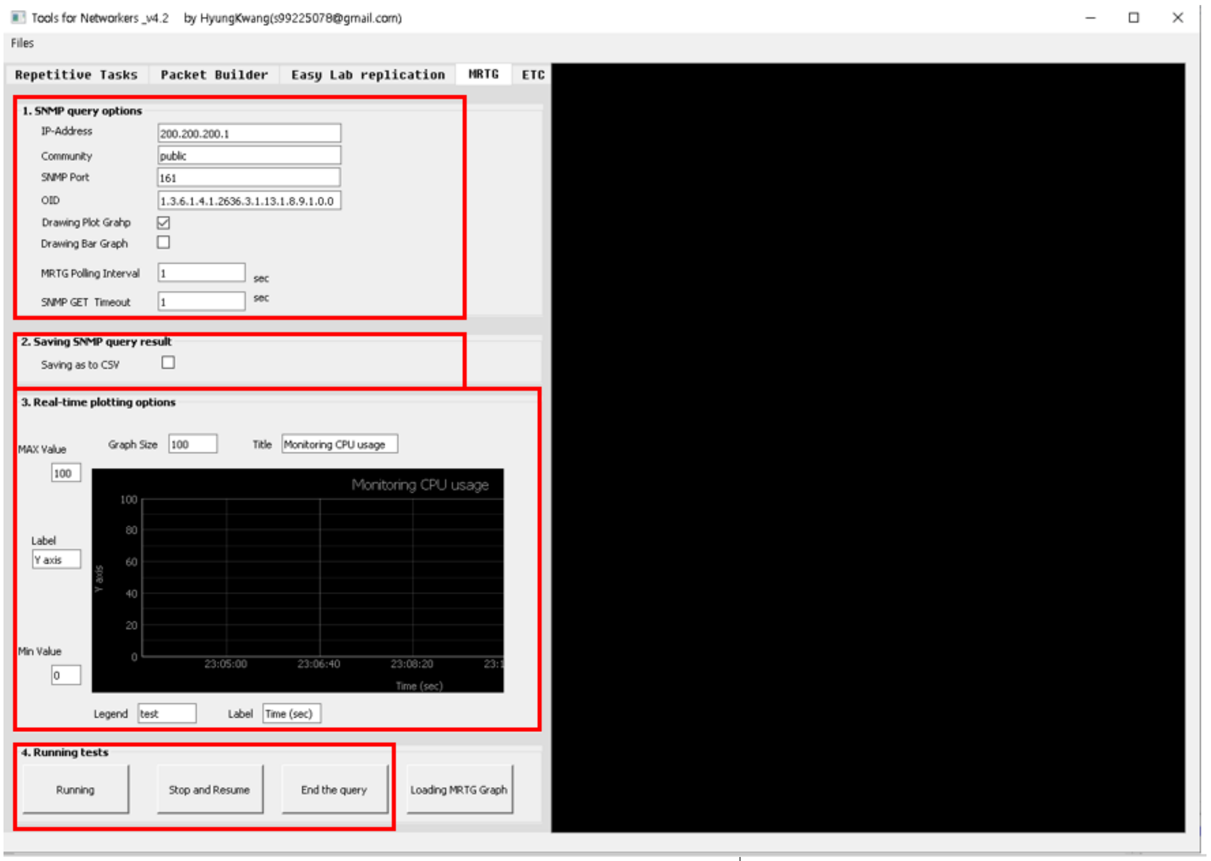
<Feature.1> "Real-time plotting" per sec.

1. **Put snmp query options properly.**
   - If you don't put options, it will not work.

2. **If you want to save OID value returned to a file, please check it. Then it will pop-up file save dialog window.**

3. **set real-time plot figure options such as) Title, figure size, legend, and so on.**
   - These are options, although you don't put anything, I will work as expected.

4. **Running/Stop/End query**
   - If you want to run multiple MRTG graph, simply put new snmp query options and click the Running button.


Tools for Networkers _v4.2 by HyungKwang(s99225078@gmail.com)

Files

Repetitive Tasks | Packet Builder | Easy Lab replication | MRTG | ETC

1. SNMP query options

IP-Address: 200.200.200.1

Community: public

SNMP Port: 161

OID: 1.3.6.1.4.1.2636.3.1.13.1.8.9.1.0.0

Drawing Plot Grahp: ✓

Drawing Bar Graph: ☐

MRTG Polling Interval: 1 sec

SNMP GET Timeout: 1 sec

2. Saving SNMP query result

Saving as to CSV: ☐

3. Real-time plotting options

MAX Value: 100

Graph Size: 100

Title: Monitoring CPU usage

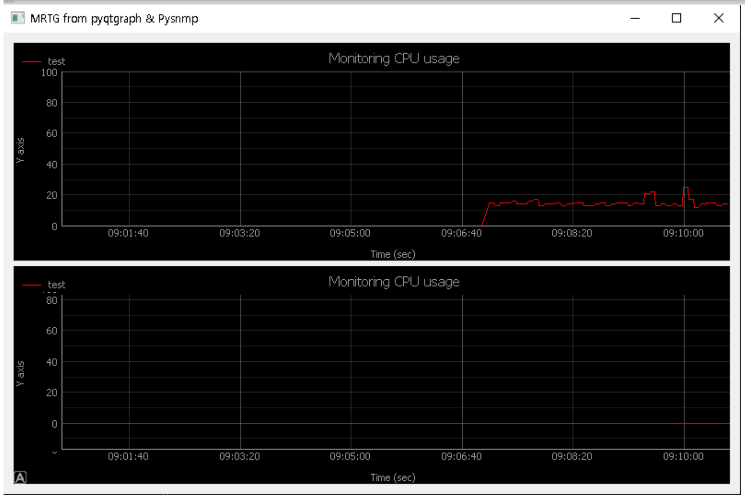
Label: Y axis

Min Value: 0

Legend: test

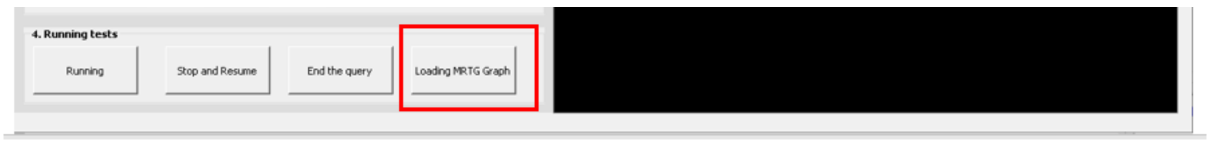
Label: Time (sec)

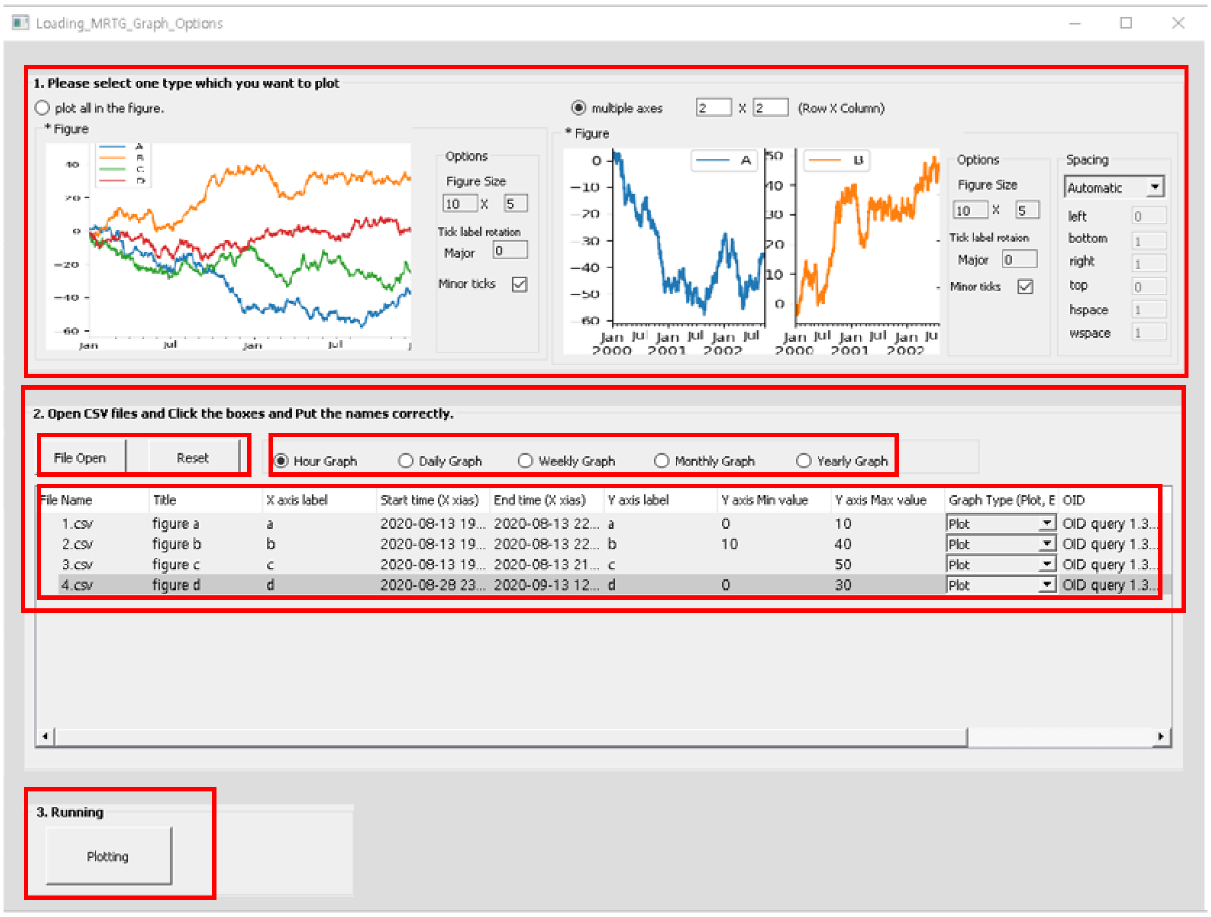
4. Running tests



<Feature.2> "Restoring figures"

To run this feature, you must click the "Loading MRTG Graph" button. It will pop-up second window.




Loading_MRTG_Graph_Options

1. Please select one type which you want to plot

( ) plot all in the figure. [✓ ● multiple axes] 2 X 2 (Row X Column)

Options: Figure Size 10 X 5; Tick label rotation Major 0; Minor ticks ✓

Options: Figure Size 10 X 5; Tick label rotaion Major 0; Minor ticks ✓

Spacing: Automatic; left 0; bottom 1; right 1; top 0; hspace 1; wspace 1

2. Open CSV files and Click the boxes and Put the names correctly.

File Open | Reset

[✓ ● Hour Graph] ( ) Daily Graph ( ) Weekly Graph ( ) Monthly Graph ( ) Yearly Graph

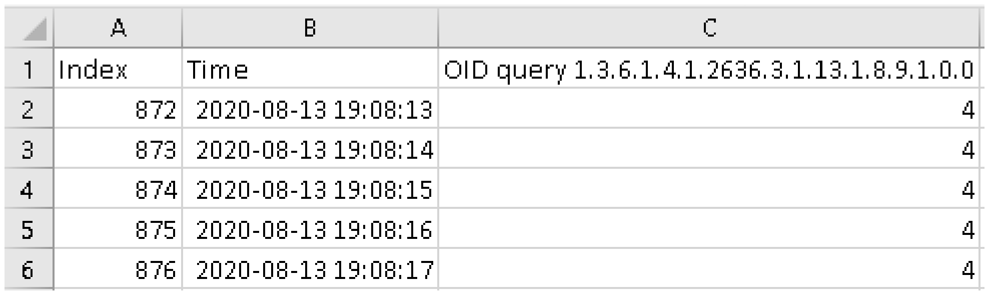
File Name	Title	X axis label	Start time (X xias)	End time (X xias)	Y axis label	Y axis Min value	Y axis Max value	Graph Type (Plot, E	OID
1.csv	figure a	a	2020-08-13 19...	2020-08-13 22...	a	0	10	Plot	OID query 1.3...
2.csv	figure b	b	2020-08-13 19...	2020-08-13 22...	b	10	40	Plot	OID query 1.3...
3.csv	figure c	c	2020-08-13 19...	2020-08-13 21...	c		50	Plot	OID query 1.3...
4.csv	figure d	d	2020-08-28 23...	2020-09-13 12...	d	0	30	Plot	OID query 1.3...

3. Running

Plotting

1. Select how you are going to plot figures. ("plot all in the figures" or "multiple axes")
- Each figure has its own options if you want you can adjust properly.

2. load CSV files you want to restore. You can select multiple files.
1) In this case CSV file data frame, it must follow this format.
There you must have header name "Index Time, OID…….", and following data.
This type of data format is from the previous "Feature.1) Real-time plotting per sec." If you run <Feature.1> "Real-time plotting" per sec, it will save data accordingly.



	A	B	C
1	Index	Time	OID query 1.3.6.1.4.1.2636.3.1.13.1.8.9.1.0.0
2	872	2020-08-13 19:08:13	4
3	873	2020-08-13 19:08:14	4
4	874	2020-08-13 19:08:15	4
5	875	2020-08-13 19:08:16	4
6	876	2020-08-13 19:08:17	4

3. set real-time plot figure options such as) Title, figure size, legend, and so on.
- These are options, although you don't put anything, I will work as expected.

2.5.3 Code review : Key functions

➢ **Tools for Networkers_v4.2.py**

```
#Used for 'MRTG'
    @QtCore.pyqtSlot() # This slot is for "Running" button in TAB-4
    def slot_13st(self):

# run_statement is using common memory to signal among multiprocess. (for 
details please refer to <3.1.3 Using Shared Memory>)
      if self.run_statement.value == 0:
         self.run_statement.value = 1

# Multiprocessing used for Asyncio pysnmp polling.
      self._multi_process_list_tab4.append(Process(target=self._snmp_pollin
g_list[self.numbering~ self.run_statement )))
      self._multi_process_list_tab4[self.numbering].start()

# Multi-threading used to draw real-time graph. Actually calling 
"class Live_plotting def plot_drawing"
      self._multi_thread_list.append(threading.Thread(name=self.numbering, 
target=self._mrtg_second_window.object_list[self.numbering].plot_drawing,~
)))
      self._multi_thread_list[self.numbering].start()

# This is a for real-time graph window.
      self._mrtg_second_window.resize(self.x_size, self.y_size)
      self._mrtg_second_window.show()
```

```
#Used for 'MRTG'
    @QtCore.pyqtSlot() # This slot is for "Stop and Resume" in TAB-4
    def slot_14st(self):

# run_statement is using common memory to signal among multiprocess. So 
when this set, def snmp_polling() while True will be out of loop 
temporally. (for details please refer to <3.1.3 Using Shared Memory>)

         if self.run_statement.value == 1:
             self.run_statement.value = 0
             print(self.run_statement.value)

         else :
             self.run_statement.value = 1
             print(self.run_statement.value)
```

```
#Used for 'MRTG'
    @QtCore.pyqtSlot() # This slot is for "End the query" in TAB-4
    def slot_15st(self):

This block does closing the second window, initialization all values used
for Tab-4, and terminating multiprocessing, multi-threading.
```

➢ **MRTG_Graph.py**

```
class SecondWindow_for_live_plotting(QWidget):

    # This is to send Closed signal to an master class object "self._mrtg_s
econd_window.Closed.connect(self.slot_15st)" in "Tools for Networkers_v4.2.
py"
    # So closing all.
   Closed = pyqtSignal()

    # This is used to detect when the close drawing 'Live_plotting' window
when the button "X" click
    def closeEvent(self, event):

# This is to add each real-time graph to "layout widget" whenever clicking
'Running' botton.
    def creat_object_add_layout(self, number, _figure_options):
        self.object_list.append(Live_plotting(_figure_options)) # calls
real-time drawing class


class Live_plotting(QWidget):

       self.pl = self.pw.plot(pen='r', name = 'Red') #This is for plot
graph
       self.pl1 = pg.BarGraphItem(x=self.x,x0=None, x1=None,y0=None,y1=None
, height=self.y, width=1, pen ='g', brush='g') # This is for bar graph

  # This is an actual drawing function thread.
    def plot_drawing(self, queue, stop_thread, saving_to_csv):

            yy = int(queue.get()) # This is most important fuction, queue
method hold thread until it gets data from
"async def async_next_snmp_request
qq.put(val)"

          # drawing plot graph.
            if self.drawing_plot:
                self.pl.setData(self.x, self.y)
          # drawing bar graph.
            if self.drawing_bar :
                self.pl1.setOpts(x=self.x,x0=None, x1=None,y0=None,y1=None,
 height=self.y, width=1, brush='g')


# This class for new window restor MRTG graph
class Second_Window_to_recall_MRTG_Graph(QWidget):

# This slot is for "File Open" button, and align all data read from csv to
treebox and merge dataframe to complete one file to draw "plot all in the
figures" or "multiple axes" later

    @QtCore.pyqtSlot()
    def slot1(self):

    fname = QFileDialog.getOpenFileNames(self, 'Open file', "", "All Files(
*);; CSV Files(*.csv)")

     #To extract DataFrame headers (OID value).
     df_Column_naming_list = list(self.df_list[0].columns)

     self.df_list.append(df)  # to merge all dataframe to complete one
dataframe.
   """
    # This is to merge all csv files opened, based on "Time" sorting.
    # 'outer' mean full merge.
    # Please refer to the link below, which written by korean language.
    # https://rfriend.tistory.com/m/258 """

    self.df_list[0] = pd.merge(self.df_list[0], self.df_list[i], how='outer
', on='Time')

# This slot is for "Plotting" button
    @QtCore.pyqtSlot()
    def slot2(self):
```

➢ **Snmp_functions.py**

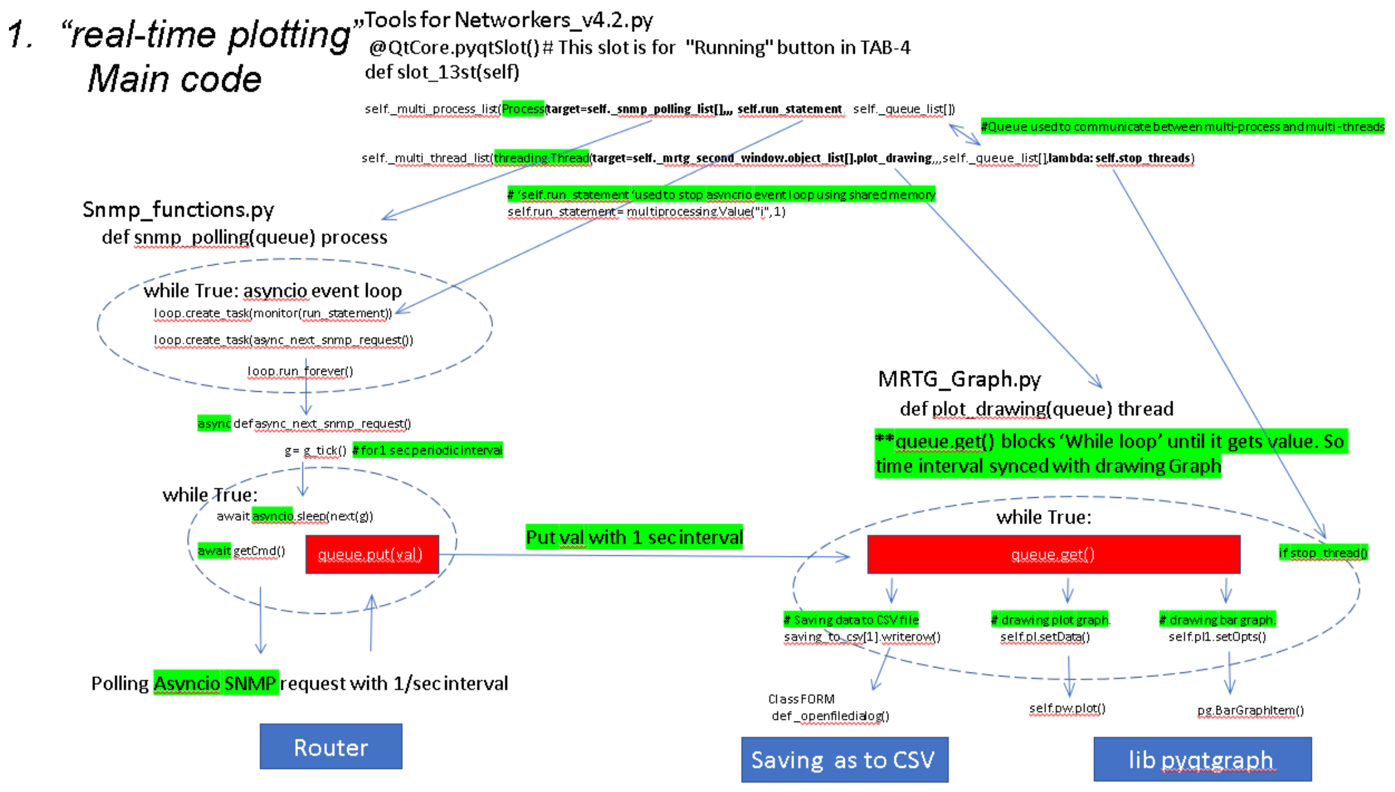
```
async def monitor(run_statement):

async def async_next_snmp_request(snmpEngine, IP_ADDRESS,COMMUNITY, OIDS, P
OLLING_INTERVAL, SNMP_GET_TIMEOUT, qq):
    """Walk SNMP oids asynchronously."""
    # await coroutine_object
    response = await getCmd(  #basic asyncio pysnmp get function.
                    )
    await asyncio.sleep(next(g)) # This is a time interval.
     qq.put(val) # Most important function

    # This is to create event loop for Asyncrio, run by multiprocessing.
“self._multi_process_list_tab4.append(Process(target=self._snmp_polling_lis
t[self.numbering~ self.run_statement ))) ”

def snmp_polling(IP_ADDRESS, COMMUNITY, OIDS, POLLING_INTERVAL,SNMP_GET_TIM
EOUT, qq, run_statement ):

  while True:

    # This is to stop the loop. Not exiting the 'While loop', but only pend
ing for a while.
     if run_statement.value == 0:
                continue
     else :
        workers = []
        workers.append(loop.create_task(monitor(run_statement))) # # This i
s to stop the loop while it's running. 'async def monitor' checks it with '
await asyncio.sleep(1)'
        workers.append(loop.create_task(async_next_snmp_request(snmpEngine,
 IP_ADDRESS, COMMUNITY, OIDS, POLLING_INTERVAL,SNMP_GET_TIMEOUT, qq ))  )

        loop.run_forever()
```

2.5.4 Architecture

For detailed figure, the PPT file(Architecture.ppt) is located under “Example” folder

2.5.4.1 real-time plotting (main code)

- multi-processing & threads & asyncio snmp interaction with queue method.

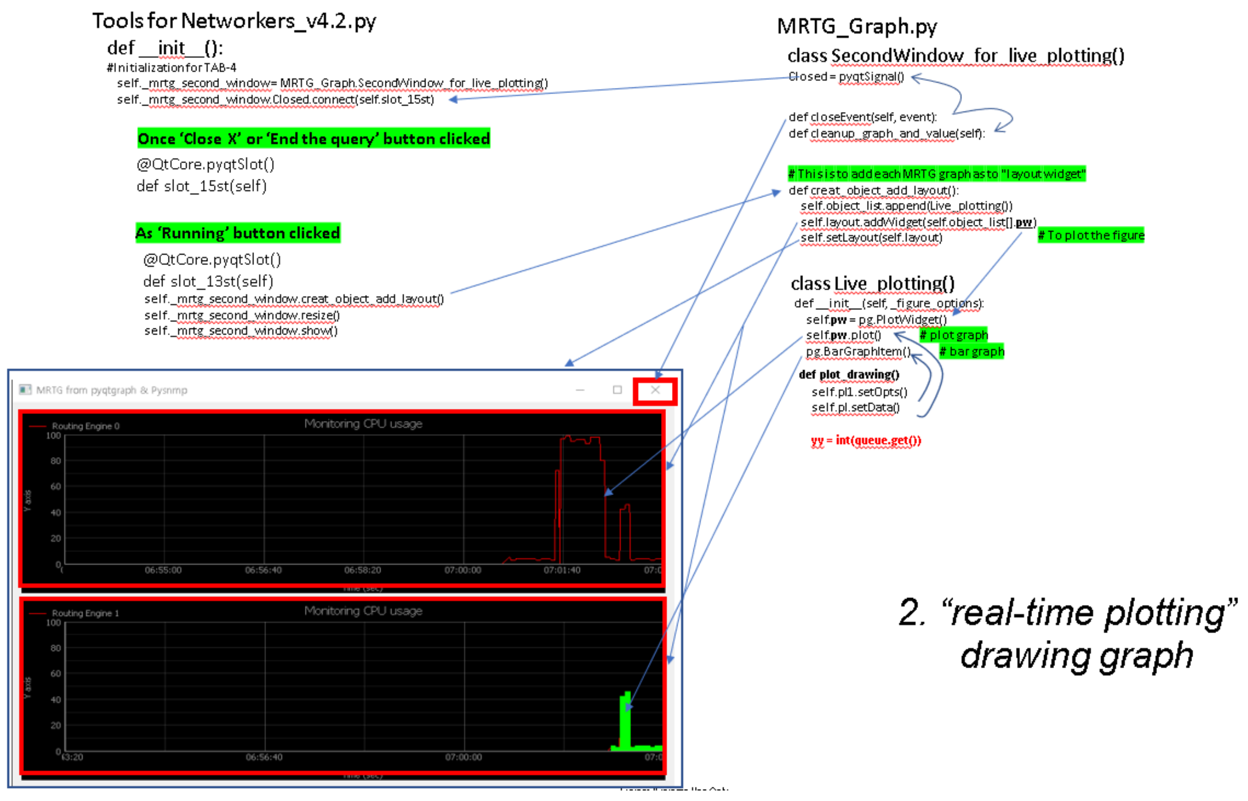

1. "real-time plotting" Main code
Snmp_functions.py
MRTG_Graph.py
Put val with 1 sec interval
Polling Asyncio SNMP request with 1/sec interval
Router
Saving as to CSV
lib pyqtgraph

```
Tools for Networkers_v4.2.py
 @QtCore.pyqtSlot() # This slot is for  "Running" button in TAB-4
def slot_13st(self)

self._multi_process_list(Process(target=self._snmp_polling_list[],,, self.run_statement  self._queue_list[])
#Queue used to communicate between multi-process and multi -threads
self._multi_thread_list(threading.Thread(target=self._mrtg_second_window.object_list[].plot_drawing,,,self._queue_list[],lambda: self.stop_threads)

# 'self.run_statement 'used to stop asyncrio event loop using shared memory
self.run_statement = multiprocessing.Value("i",1)
```

```
Snmp_functions.py
 def snmp_polling(queue) process

while True: asyncio event loop
 loop.create_task(monitor(run_statement))
 loop.create_task(async_next_snmp_request())
 loop.run_forever()

async def async_next_snmp_request()
 g = g_tick()  # for1 sec periodic interval
while True:
 await asyncio.sleep(next(g))
 await getCmd()
 queue.put(val)
```

```
MRTG_Graph.py
 def plot_drawing(queue) thread
**queue.get() blocks 'While loop' until it gets value. So time interval synced with drawing Graph

while True:
 queue.get()
 if stop_thread()
 # Saving data to CSV file
 saving_to_csv[1].writerow()
 # drawing plot graph.
 self.pl.setData()
 # drawing bar graph.
 self.pl1.setOpts()

Class FORM
 def _openfiledialog()
self.pw.plot()
pg.BarGraphItem()
```

2.5.4.2 real-time plotting (drawing graph) by pyqtgraph lib



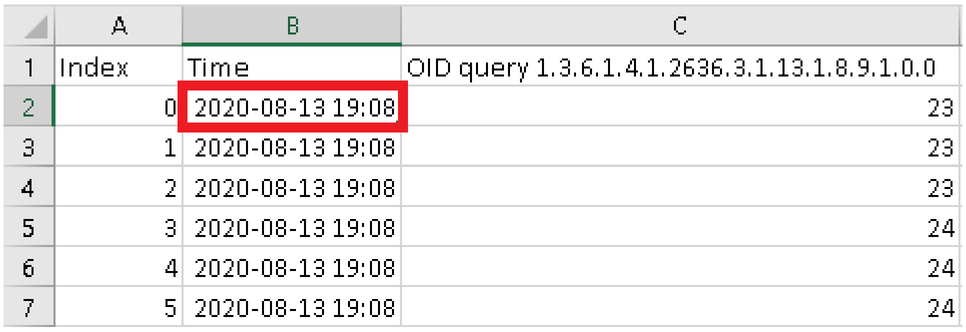
```
Tools for Networkers_v4.2.py
 def __init__():
 #Initialization for TAB-4
 self._mrtg_second_window = MRTG_Graph.SecondWindow_for_live_plotting()
 self._mrtg_second_window.Closed.connect(self.slot_15st)

Once 'Close X' or 'End the query' button clicked
@QtCore.pyqtSlot()
def slot_15st(self)

As 'Running' button clicked
 @QtCore.pyqtSlot()
 def slot_13st(self)
 self._mrtg_second_window.creat_object_add_layout()
 self._mrtg_second_window.resize()
 self._mrtg_second_window.show()
```

```
MRTG_Graph.py
 class SecondWindow_for_live_plotting()
 Closed = pyqtSignal()

 def closeEvent(self, event):
 def cleanup_graph_and_value(self):

 # This is to add each MRTG graph as to "layout widget"
 def creat_object_add_layout():
   self.object_list.append(Live_plotting())
   self.layout.addWidget(self.object_list[].pw)   # To plot the figure
   self.setLayout(self.layout)

 class Live_plotting()
  def __init__(self, _figure_options):
    self.pw = pg.PlotWidget()
    self.pw.plot()   # plot graph
    pg.BarGraphItem()   # bar graph
   def plot_drawing()
     self.pl1.setOpts()
     self.pl.setData()

     yy = int(queue.get())
```

Non-Juniper

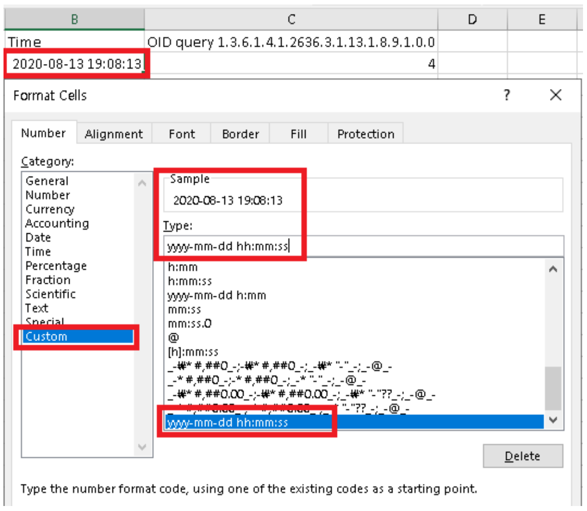
2.5.5 ETC

- While you are running **<Feature.1> "Real-time plotting" per sec, once you save csv file option, if you open a scv file, you can see 'Time' data like below.**
Actually it's save on per sec. so to see the complete time data, please follow below.



	A	B	C
1	Index	Time	OID query 1.3.6.1.4.1.2636.3.1.13.1.8.9.1.0.0
2	0	2020-08-13 19:08	23
3	1	2020-08-13 19:08	23
4	2	2020-08-13 19:08	23
5	3	2020-08-13 19:08	24
6	4	2020-08-13 19:08	24
7	5	2020-08-13 19:08	24

➢ **please change the format cell type.**



B	C	D	E
Time	OID query 1.3.6.1.4.1.2636.3.1.13.1.8.9.1.0.0		
2020-08-13 19:08:13	4		

Format Cells

Number | Alignment | Font | Border | Fill | Protection

Category:
General
Number
Currency
Accounting
Date
Time
Percentage
Fraction
Scientific
Text
Special
Custom

Sample
2020-08-13 19:08:13

Type:
yyyy-mm-dd hh:mm:ss

```
h:mm
h:mm:ss
yyyy-mm-dd h:mm
mm:ss
mm:ss.0
@
[h]:mm:ss
_-₩* #,##0_-;-₩* #,##0_-;_-₩* "-"_-;_-@_-
_-* #,##0_-;-* #,##0_-;_-* "-"_-;_-@_-
_-₩* #,##0.00_-;-₩* #,##0.00_-;_-₩* "-"??_-;_-@_-
yyyy-mm-dd hh:mm:ss
```

Delete

Type the number format code, using one of the existing codes as a starting point.

3. Library & Function & Useful code example used in this tool

****All examples introduced here, are located under "Example" folder**

3.1 Python3.8

3.1.1 How to capture Exception and pop it up messages.

In TAB-2. When you execute 'running', SCAPY raises an error due to PCAP is not running in root privilege. So, once SCAPY lib raises an error, this could be captured with the exception, and popup.

```
Traceback (most recent call last):
~
    raise Scapy_Exception(
scapy.error.Scapy_Exception: Interface is invalid (no pcap match found) !
```

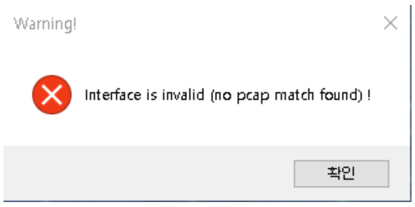
➢ **From the code "Tools for Networkers_v4.2.py"**

```
import ctypes    # This default Python lib. You don't need to install it.

 except Exception as e:
                a = str(e)
                ctypes.windll.user32.MessageBoxW(0, a , "Warning!", 16)
                pass
```

➢ **'cytpes' MessageBoxW Options**

```
* Reference : 'ctypes'
   - https://stackoverflow.com/questions/2963263/how-can-i-create-a-
simple-message-box-in-python

   Button styles:
    0 : OK
    1 : OK | Cancel
    2 : Abort | Retry | Ignore
    3 : Yes | No | Cancel
    4 : Yes | No
    5 : Retry | No
    6 : Cancel | Try Again | Continue

   To also change icon, add these values to previous number
    16 Stop-sign icon
    32 Question-mark icon
    48 Exclamation-point icon
    64 Information-sign icon consisting of an 'i' in a circle
```

➢ **'ctypes' Pop-up messages**


Warning!

Interface is invalid (no pcap match found) !

3.1.2 Running shell commands

If you want to run python script directly in Juniper product(FreeBSD) or Linux box, your python commands have to run on Shell such as) Csh, bash…or else.

So for this, you have to use subprocess lib so that you are able to run shell commands

- **"python_Executing Shell commands in Junos.py"**

```
from __future__ import division
import sys
import re
import getopt
import subprocess   #This used for shell script

#cmd = "cprod -A fpc3 -c "show syslog messages""
lines = []

lines = subprocess.check_output("cprod -A fpc3 -c \"show syslog messages\"", shell=True)
     # <==== subprocess runs on SHELL..so with the command you can run shell, c, C++ or else o
n shell
##check_output used to collect the outputs executed by subprocess
print (lines)
```

3.1.3 Using shared memory

```
* Reference : python shared memory
  - https://docs.python.org/3/library/multiprocessing.shared_memory.html

--snip—-

* Sharing state between processes

if you really do need to use some shared data then multiprocessing provides
a couple of ways of doing so.

* Shared memory
Data can be stored in a shared memory map using Value or Array. For
example, the following code. The 'd' and 'i' arguments used when creating
num and arr are typecodes of the kind used by the array module: 'd'
indicates a double precision float and 'i' indicates a signed integer.
```

- **"shared_memory.py"**

```

from multiprocessing import Process, Value, Array

def f(n, a):
    n.value = 3.1415927
    for i in range(len(a)):
        a[i] = -a[i]

if __name__ == '__main__':
    num = Value('d', 0.0)
    arr = Array('i', range(10))

    p = Process(target=f, args=(num, arr))
    p.start()
    p.join()

    print(num.value)
    print(arr[:])

3.1415927
[0, -1, -2, -3, -4, -5, -6, -7, -8, -9]

# Race condition
"""import time
from multiprocessing import Process, Value

def func(val):
    for i in range(50):
        time.sleep(0.01)
        val.value += 1

if __name__ == '__main__':
    v = Value('i', 0)
    procs = [Process(target=func, args=(v,)) for i in range(10)]
    for p in procs: p.start()
    for p in procs: p.join()
    print (v.value)"""

379

#After fixing Race-condition with Lock

import time
from multiprocessing import Process, Value, Lock

def func(val, lock):
    for i in range(50):
        time.sleep(0.01)
        with lock:
            val.value += 1

if __name__ == '__main__':
    v = Value('i', 0)
    lock = Lock()
    procs = [Process(target=func, args=(v,lock)) for i in range(10)]
    for p in procs: p.start()
    for p in procs: p.join()
    print (v.value)
500
```

3.1.4 How to extract only file name

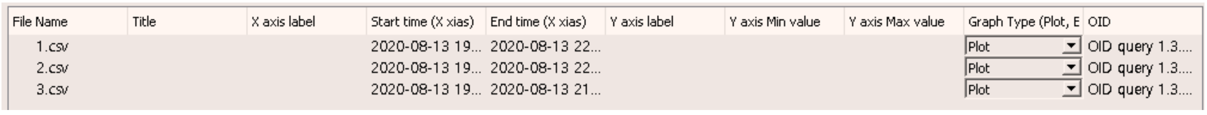
```
from os.path import basename
filename = 'C:\\temp\\test\\text.txt'

print(basename(filename))
text.txt
```

3.1.5 How to add Dictionary list value.

There are 2 methods to add Dictionary list value.

1. **dictionary.setdefault(keyname, value)**
2. **dict1[i] = [x, x, x]**

- **from "Loading MRTG Graph"**



File Name	Title	X axis label	Start time (X xias)	End time (X xias)	Y axis label	Y axis Min value	Y axis Max value	Graph Type (Plot, E	OID
1.csv			2020-08-13 19...	2020-08-13 22...				Plot	OID query 1.3....
2.csv			2020-08-13 19...	2020-08-13 22...				Plot	OID query 1.3....
3.csv			2020-08-13 19...	2020-08-13 21...				Plot	OID query 1.3....

- **from "python_dictionary_list.py"**

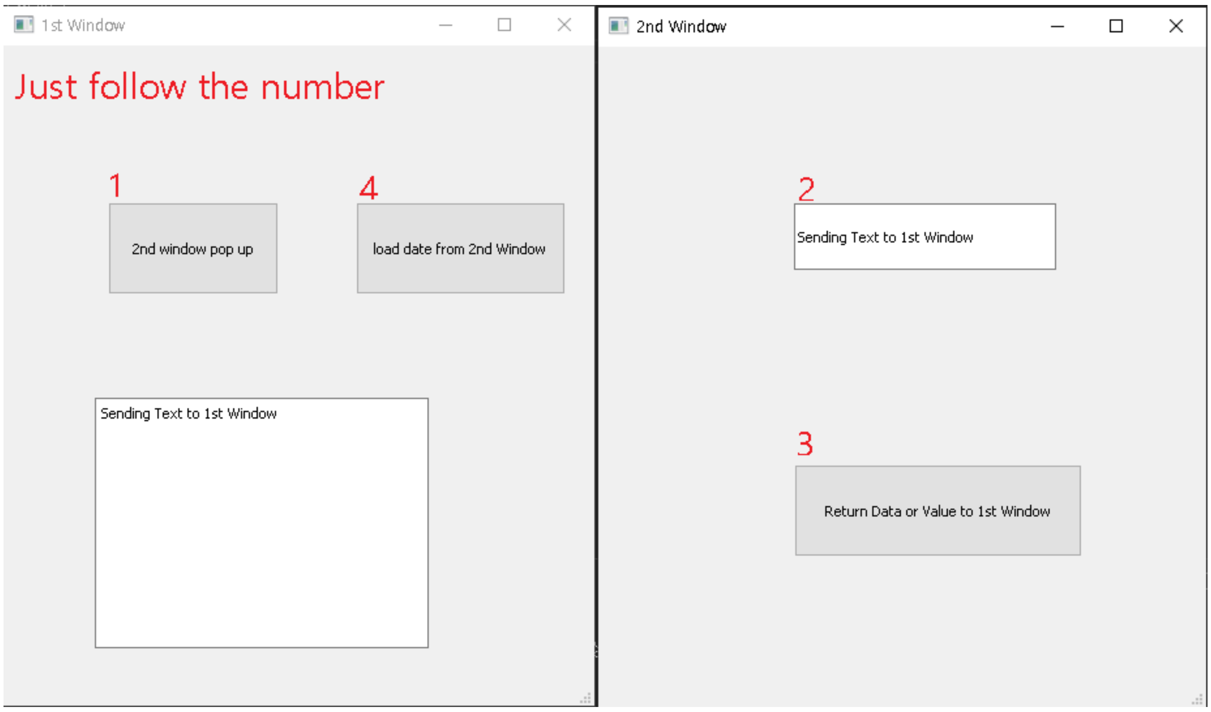
```
  plot_item_dict = {}

  for i in range(child_count):
  # item has each QTreeWidgetItem while it's running on 'for' statement.
      item = root.child(i)
     #Below is to take DropBox 5, 11 items value chosen into dictionary.
      for j in range(12):
        if j == 8 or j == 11:
          qComboBox = self.mytree.itemWidget(self.mytree_item_list[i], j)
          text_selected = qComboBox.currentText()
           plot_item_dict.setdefault(i, []).append(text_selected)
         else :
        plot_item_dict.setdefault(i, []).append(item.text(j))
  print(plot_item_dict)

{0: ['1.csv', '', '', '2020-08-13 19:08:13', '2020-08-13 22:15:00', '', '', '',
'Plot', 'OID query 1.3.6.1.4., '', 'blue'],
 1: ['2.csv', '', '', '2020-08-13 19:08:13', '2020-08-13 22:15:00', '', '', '',
'Plot', 'OID query 1.3.6.1.4', '', 'blue'],
 2: ['3.csv', '', '', '2020-08-13 19:08:13', '2020-08-13 21:44:33', '', '', '',
'Plot', 'OID query 1.3.6.1.4., '', 'blue']}

############# From 'python_dictionary_list.py'#########################
car = {
  "brand": "Ford",
  "model": "Mustang",
   "year": 1964
}
x = car.setdefault("color",  "white")
print(x)
white
#############
dict1 = {}
x = 0
for i in range(5) :
    x = x + i
    dict1[i] = [x, x, x]
print(dict1)
{0: [0, 0, 0], 1: [1, 1, 1], 2: [3, 3, 3], 3: [6, 6, 6], 4: [10, 10, 10]}
```

3.2 Pyqt5

3.2.1 How to crate/communicate between main and second window.

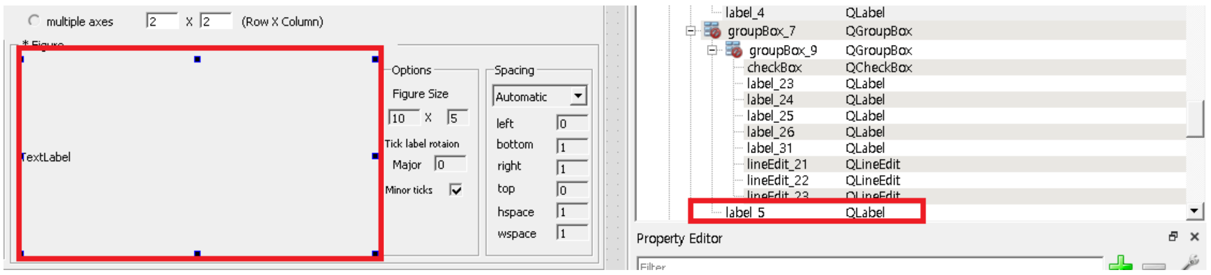
- **from “Examples\second window\master.py”**

**** I created window and did slot link on “QT Designer”, for this, you have to install QT designer.**


1st Window
Just follow the number
1
2nd window pop up
4
load date from 2nd Window
Sending Text to 1st Window
2nd Window
2
Sending Text to 1st Window
3
Return Data or Value to 1st Window

3.2.2 How to add pictures in UI (using QT designer)

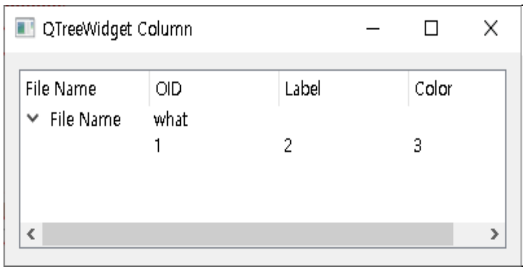
1. set QLabel properly.
2. Then code like below.(For the QPixmap, you have to use label)



```
from PyQt5.QtGui import  QPixmap


self.label_4.setPixmap(QPixmap(os.path.join(self.root, './Images/plot2.png')).scaled(340,200))
self.label_5.setPixmap(QPixmap(os.path.join(self.root, './Images/plot3.png')).scaled(440,200))
```

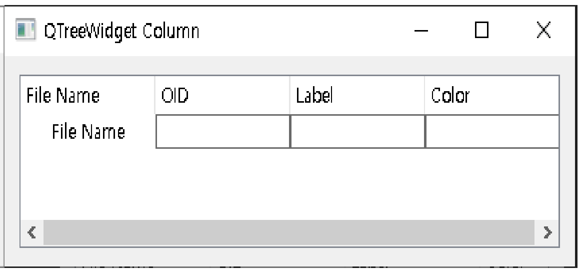
3.2.3 QtreeWidget

1). "PYQT5_QtreeWidget1.py"



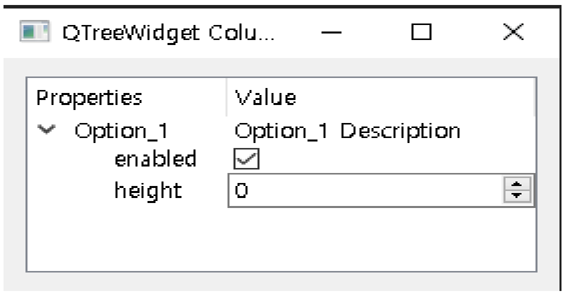
File Name	OID	Label	Color
File Name	what		
	1	2	3

2). "PYQT5_QtreeWidget2.py"



File Name	OID	Label	Color
File Name			

3). "PYQT5_QtreeWidget3.py"



Properties	Value
Option_1	Option_1 Description
enabled	✓
height	0

3.2.4 Click event handling (checkBox, pushButton, comboBox)

1. checkBox state change

```
 self.checkBox_4.stateChanged.connect(self.checkBox_state_changed)

def checkBox_state_changed(self):
    if self.checkBox_4.isChecked():
        print("CHECKED!")
    else:
        print("UNCHECKED!")
```

2. pushButton clicked

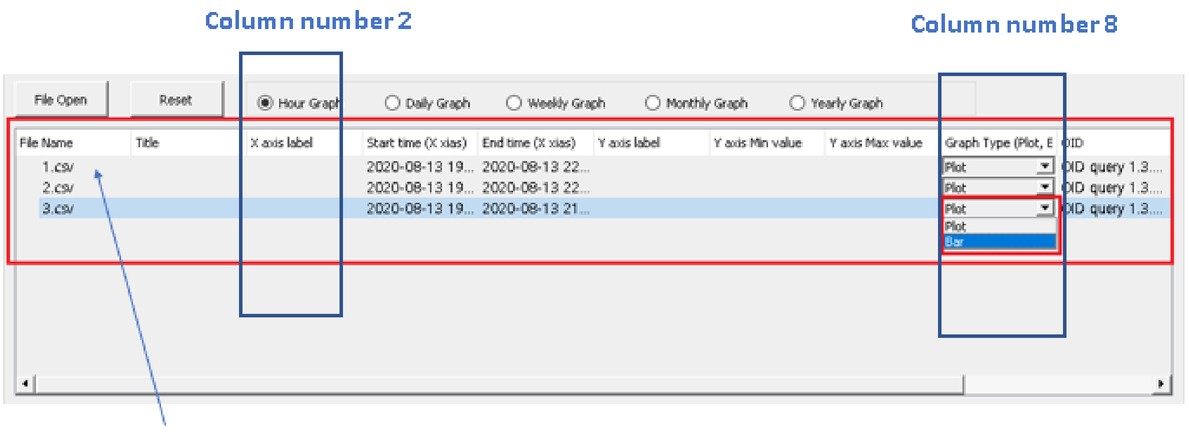
```
 self.pushButton.clicked.connect(self.popup_new_window)

def popup_new_window(self):
    self.otherview = MRTG_Graph.Second_Window_to_recall_MRTG_Graph()
    self.otherview.show()
```

3. **comboBox currentTextChanged changed.**

```
 self.comboBox.currentTextChanged.connect(self._changing_lineEdit_status)

 def _changing_lineEdit_status(self):
        if self.comboBox.currentText() == 'Automatic' :
          self.lineEdit_3.setEnabled(False)
        else :
          self.lineEdit_3.setEnabled(True)
```

3.2.5 How to set comboBox data and read it on Qtreewidget

```
# setting/initializing comboBox onto QtreeWidget

self.combo_box_graph_types = QComboBox(self.mytree)
self.combo_box_graph_types.addItem('Plot')
elf.combo_box_graph_types.addItem('Bar')
self.mytree.setItemWidget(self.mytree_item_list[x], 8, self.combo_box_graph
_types) # '8' is column number.
self.mytree_item_list[x] is row list

# reading current comboBox data selected from QtreeWidget
for j in range(12):
   if j == 8 or j == 11:
       qComboBox = self.mytree.itemWidget(self.mytree_item_list[i], j)
       text_selected = qComboBox.currentText()
       plot_item_dict.setdefault(i, []).append(text_selected)
```



File Name	Title	X axis label	Start time (X xias)	End time (X xias)	Y axis label	Y axis Min value	Y axis Max value	Graph Type (Plot, E	OID
1.csv			2020-08-13 19...	2020-08-13 22...				Plot	OID query 1.3....
2.csv			2020-08-13 19...	2020-08-13 22...				Plot	OID query 1.3....
3.csv			2020-08-13 19...	2020-08-13 21...				Plot	OID query 1.3....

self.mytree_item_list[x] is a "row" list

#Below is a way to read comboBox current text from QtreeWidget

```
qComboBox= self.mytree.itemWidget(self.mytree_item_list[i],j)
text_selected = qComboBox.currentText()
```

3.2.6 How to change second window size

There are 2 methods from my code.

1. Using the function ‘Resize’

```
    self._mrtg_second_window.resize(self.x_size, self.y_size)
    self._mrtg_second_window.show()
```

2. using the function ‘setFixedWidth’ and ‘setFixedHeight’
   To use 'setFixedWidth', 'setFixedHeight', you must initialize both method under "class SecondWindow_for_live_plotting"

```
          self._mrtg_second_window.setFixedWidth(self.x_size)
          self._mrtg_second_window.setFixedHeight(self.y_size)

class SecondWindow_for_live_plotting(QWidget):
           initial_width = 800
          # setting  the fixed width of window
          self.setFixedWidth(initial_width)

          initial_height = 250
          # setting  the fixed height of window
          self.setFixedHeight(initial_height)
```

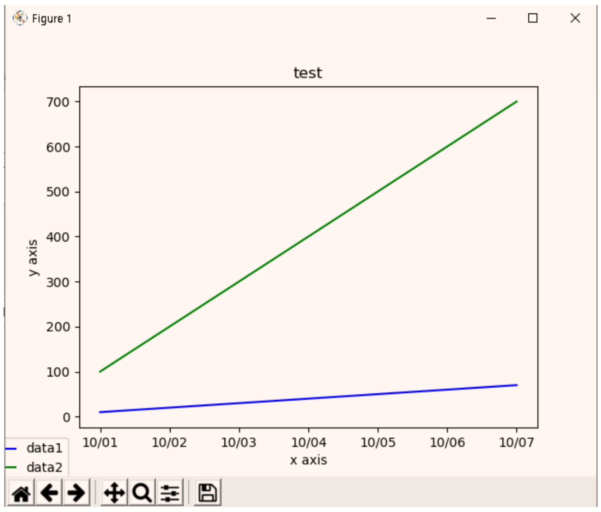
3.3 Matplotlib

3.3.1 Using pandas DataFrame in matplotlib.

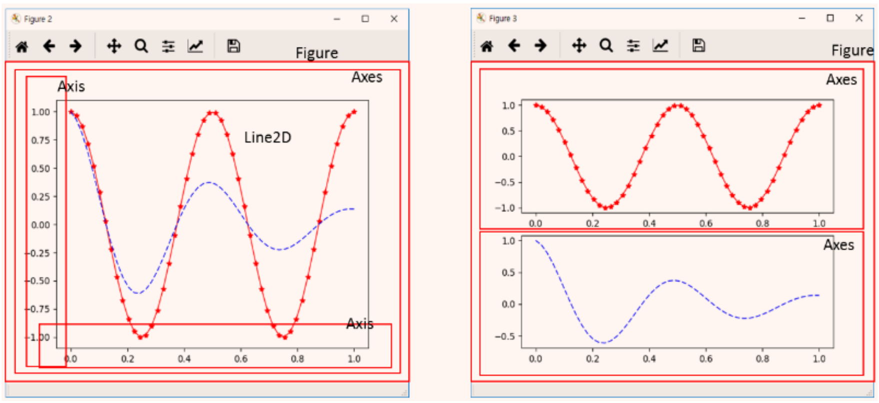
- **matplotlib_with_pandas.py**

```
import io
import pandas as pd
import matplotlib.pyplot as plt
import matplotlib.dates as mdates


data = """
year,data1,data2
2020-10-01,10,100
2020-10-02,20,200
2020-10-03,30,300
2020-10-04,40,400
2020-10-05,50,500
2020-10-06,60,600
2020-10-07,70,700
"""
df = pd.read_csv(io.StringIO(data))
print(df)

        year  data1  data2
0  2020-10-01     10    100
1  2020-10-02     20    200
2  2020-10-03     30    300
3  2020-10-04     40    400
4  2020-10-05     50    500
5  2020-10-06     60    600
6  2020-10-07     70    700

df = pd.read_csv(io.StringIO(data),index_col=0)
df.index = pd.to_datetime(df.index)
print(df)

          data1  data2
year
2020-10-01     10    100
2020-10-02     20    200
2020-10-03     30    300
2020-10-04     40    400
2020-10-05     50    500
2020-10-06     60    600
2020-10-07     70    700
```

```
plt.gca().xaxis.set_major_formatter(mdates.DateFormatter('%m/%d'))
line1, = plt.plot(df.index, df['data1'], color='b')
line2, = plt.plot(df.index, df['data2'], color='g')
line1.set_label('data1')
line2.set_label('data2')

plt.legend(bbox_to_anchor=(-0.01,-0.01))
plt.title('test')
plt.ylabel("y axis")
plt.xlabel("x axis")
plt.show()
```



3.3.2 How are different among figure, Axis and Axes.



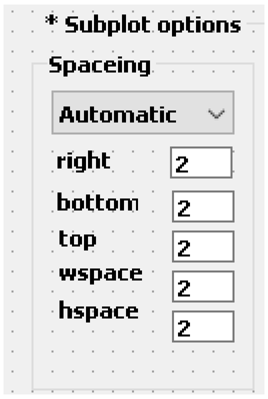
3.3.3 subplots_adjust

```
subplot_adjusts used to adjust spacing among subplots.

matplotlib.pyplot.subplots_adjust(left=None, bottom=None, right=None,
top=None, wspace=None, hspace=None)
```

```
left = 0.125  : the left side of the subplots of the figure
right = 0.9   : the right side of the subplots of the figure
bottom = 0.1  : the bottom of the subplots of the figure
top = 0.9     # the top of the subplots of the figure
wspace = 0.2  # the amount of width reserved for space between subplots,
           # expressed as a fraction of the average axis width
hspace = 0.2  # the amount of height reserved for space between subplots,
           # expressed as a fraction of the average axis height
```

- **From my code. (MRTG_Graph.py)**

```
if self.comboBox.currentText() == 'Automatic' :
     plt.tight_layout()

if self.comboBox.currentText() == 'Manual' :
     plt.subplots_adjust(left = int(self.lineEdit_3.text()), bottom =
int(self.lineEdit_4.text()), right = int(self.lineEdit_5.text()), top =
int(self.lineEdit_6.text()), hspace = int(self.lineEdit_7.text()), wspace =
int(self.lineEdit_8.text()))
```


* Subplot options

Spaceing

Automatic

right 2

bottom 2

top 2

wspace 2

hspace 2

3.4 Pandas

3.4.1 DataFrame, Series creation, basic statement (iloc, loc)

- **“pandas_dataframe_loc_iloc_example.py”**

```
import pandas as pd
import numpy as np
import io

######## DataFrame Creation example ############

test_2D_array = np.array([[1, 2, 3], [4, 5, 6]])
df = pd.DataFrame(test_2D_array)
print(df)
   0  1  2
0  1  2  3
1  4  5  6

df = pd.DataFrame(np.arange(16).reshape(4,4), index=['test1', 'test2','test
3','test4'], columns=['one','two','three','four'])
print(df)

      one  two  three  four
test1    0    1      2     3
test2    4    5      6     7
test3    8    9     10    11
test4   12   13     14    15

df = pd.DataFrame(np.arange(16).reshape(4,4), index=range(0,4), columns=['o
ne','two','three','four'])
print(df)

   one  two  three  four
0    0    1      2     3
1    4    5      6     7
2    8    9     10    11
3   12   13     14    15

test_dict = {"one": ['1', '2','3', '4'], "two": ['5', '6','7', '8'], "three
": ['9', '10','11','12']}
df = pd.DataFrame(test_dict)
print(df)
  one two three
0   1   5     9
1   2   6    10
2   3   7    11
```

```
3   4   8    12


data = """
year,data1,data2
2020-10-01,10,100
2020-10-02,20,200
2020-10-03,30,300
2020-10-04,40,400
2020-10-05,50,500
2020-10-06,60,600
2020-10-07,70,700
"""
df = pd.read_csv(io.StringIO(data))
print(df)

df = pd.read_csv(io.StringIO(data),index_col=0)
df.index = pd.to_datetime(df.index)
print(df)
         year  data1  data2
0  2020-10-01     10    100
1  2020-10-02     20    200
2  2020-10-03     30    300
3  2020-10-04     40    400
4  2020-10-05     50    500
5  2020-10-06     60    600
6  2020-10-07     70    700
            data1  data2
year
2020-10-01     10    100
2020-10-02     20    200
2020-10-03     30    300
2020-10-04     40    400
2020-10-05     50    500
2020-10-06     60    600
2020-10-07     70    700

######## Series Creation example ###############
df = pd.Series({"one":"aa", "two":"bb", "three":"cc", "four":"dd"})
print(df)
one      aa
two      bb
three     cc
four     dd
dtype: object
####### Test ###################################

df = pd.DataFrame({"one":[1,4,7], "two":[2,5,8], "three":[3,6,9]})
```

```
print(df)
   one  two  three
0    1    2      3
1    4    5      6
2    7    8      9

#  `iloc[]` used for a row
print(df.iloc[0])
one      1
two      2
three    3
Name: 0, dtype: int64
print(df.loc[0])
one      1
two      2
three    3
Name: 0, dtype: int64

#  `loc[]` used for a column
print(df.loc[:,'one'])
0    1
1    4
2    7
Name: one, dtype: int64

print(df['one'])
0    1
1    4
2    7
Name: one, dtype: int64
# specific row & column
print(df.loc[0]['two']) #please figure out how it differs
2
#print(df.loc[-1]['B'])  #<===== This return's an error

# "-1" mean last index
print(df.iloc[[0],[1]])
   two
0    2

print(df.iloc[[-1],[1]])
   two
2    8
print(df.iloc[0],['three'])
one      1
two      2
three    3
Name: 0, dtype: int64 ['three']
```

```
print(df.iloc[-1],['three'])
one      7
two      8
three    9
Name: 2, dtype: int64 ['three']

print(df.iloc[:,1])
0    2
1    5
2    8
Name: two, dtype: int64
print(df.iloc[:,1:2])
   two
0    2
1    5
2    8
print(df.iloc[:,[0,2]])
   one  three
0    1      3
1    4      6
2    7      9
print(df.iloc[[0,-1],:])
   one  two  three
0    1    2      3
2    7    8      9
```

3.4.2 drop, set_index

```
 set_index : used to change index.
 drop : used to drop column, row

 DataFrame.drop(labels=None, axis=0, index=None, columns=None, level=None,
inplace=False)

 labels = 'index' or 'column' name
 axis = 0 (index), 1(column)
 inplace : 'True' means updated dropped value to Dataframe.
           'False' returns with label dropped.
```

- **"pandas_drop_set_index.py"**

```
import pandas as pd

test_dict = {"one": ['1', '2','3', '4'], "two": ['5', '6','7', '8'], "three
": ['9', '10','11','12']}

df = pd.DataFrame(test_dict)
print(df)
```

```
  one  two  three
0   1   5     9
1   2   6    10
2   3   7    11
3   4   8    12

df.drop('two',axis=1 )
print(df)
  one  two  three
0   1   5     9
1   2   6    10
2   3   7    11
3   4   8    12

df.drop('two',axis=1, inplace=True)
print(df)
  one  three
0   1     9
1   2    10
2   3    11
3   4    12

df.drop(2,axis=0, inplace=True)
print(df)
  one  three
0   1     9
1   2    10
3   4    12

df.set_index('three', inplace=True)
print(df)

      one
three
9       1
10      2
12      4
```

3.4.3 How to change Dataframe to list

- **"pandas_to_change_dateframe_to_list.py"**

```
import pandas as pd

df = pd.DataFrame({"one":[1,4,7], "two":[2,5,8], "three":[3,6,9]})

print(df)
```

```
   one  two  three
0    1    2      3
1    4    5      6
2    7    8      9

print(df['two'])
0    2
1    5
2    8
Name: two, dtype: int64

a = list(df['two'].tolist())
print(a)
[2, 5, 8]

x = df.iloc[:,1]
x = list(x.tolist())
print(x)
[2, 5, 8]
```

3.4.4 How to change index as to datetime.

```
When you set an index or use an index with date-time,
pandas doesn't work well with custom date-time formats.
So you have to use this statement

Ex)
pd.to_datetime(df['Time'], format='%Y-%m-%d %H:%M:%S', errors='raise')
df.index = pd.to_datetime(df.index)
```

- **"pandas_indexing_datetime.py"**

```
import pandas as pd

test_dict = {"numbering": ['872', '873','874', '875'], "Time": ['2020-08-13 19:08:13', '2020-08-13 19:08:14','2020-08-13 19:08:15', '2020-08-13 19:08:16'], "OID_number": ['7', '8','9','10']}

df = pd.DataFrame(test_dict)
print(df)

  numbering                 Time OID_number
0       872  2020-08-13 19:08:13          7
1       873  2020-08-13 19:08:14          8
2       874  2020-08-13 19:08:15          9
3       875  2020-08-13 19:08:16         10
```

```
#pandas just doesn't work well with custom date-
time formats, <class 'pandas.core.indexes.datetimes.DatetimeIndex'>
df['Time'] = pd.to_datetime(df['Time'], format='%Y-%m-%d %H:%M:%S', errors=
'raise')
df.drop('numbering',axis=1, inplace=True) #"axis=1, inplace=True" means, pl
ease refer to previous explanation.
df.set_index('Time', inplace=True)
print(df)
                  OID_number
Time
2020-08-13 19:08:13         7
2020-08-13 19:08:14         8
2020-08-13 19:08:15         9
2020-08-13 19:08:16        10
```

3.4.5 DataFrame merge

- **"pandas_merge.py"**
- **Reference : pandas dataframe merge**
  - https://rfriend.tistory.com/m/258

```
import pandas as pd

test1_dict = { "Time": ['2020-08-13 19:08:13', '2020-08-13 19:08:14',
'2020-08-13 19:08:15', '2020-08-13 19:08:16'], "OID": ['7', '7','7','7']}

test2_dict = { "Time": ['2020-08-13 19:08:15', '2020-08-13 19:08:16',
'2020-08-13 19:08:17', '2020-08-13 19:08:18'], "OID":['10','10','10','10']}


df1 = pd.DataFrame(test1_dict)
print(df1)
                Time OID
0  2020-08-13 19:08:13   7
1  2020-08-13 19:08:14   7
2  2020-08-13 19:08:15   7
3  2020-08-13 19:08:16   7

df2 = pd.DataFrame(test2_dict)
print(df2)
                Time OID
0  2020-08-13 19:08:15  10
1  2020-08-13 19:08:16  10
2  2020-08-13 19:08:17  10
3  2020-08-13 19:08:18  10

df = pd.merge(df1, df2, how='outer', on='Time') # 'outer' mean full merge.
```

```
print(df)

                 Time OID_x OID_y
0  2020-08-13 19:08:13     7   NaN
1  2020-08-13 19:08:14     7   NaN
2  2020-08-13 19:08:15     7    10
3  2020-08-13 19:08:16     7    10
4  2020-08-13 19:08:17   NaN    10
5  2020-08-13 19:08:18   NaN    10
```

3.4.6 Pandas functions etc

1. **Extracting header info. (df.columns)**

➢ **"pandas_functions_etc.py"**

```
import pandas as pd

test_dict = { "Time": ['2020-08-13 19:08:13', '2020-08-13 19:08:14',
'2020-08-13 19:08:15', '2020-08-13 19:08:16'], "OID": ['7', '7','7','7']}

df = pd.DataFrame(test_dict)

# 1). Extracting header info
print(df.columns)

Index(['Time', 'OID'], dtype='object')
print(list(df.columns))

['Time', 'OID']
```

4. Errors

4.1 Error.1

If you enable Bar graph legnd, you will face error, because 'Bar graph legnd' doesn't work well at this version. P lease do not use "legend.addItem"

```
legend.addItem(self.pl1, 'yyy')

Traceback (most recent call last):
  File "C:\Python38\lib\site-
packages\pyqtgraph\graphicsItems\LegendItem.py", line 238, in paint
    if opts['antialias']:
KeyError: 'antialias'
```

4.2 Error.2

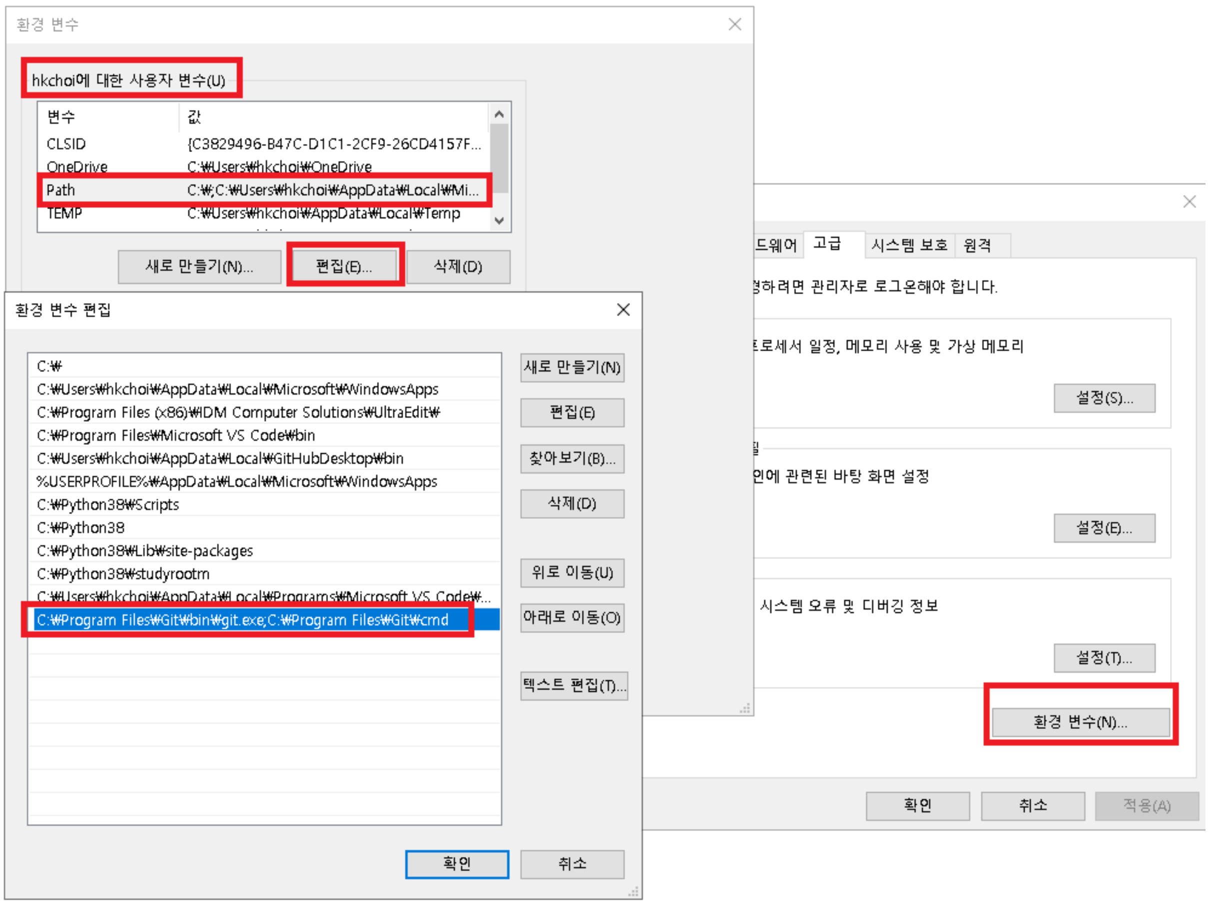
After you installed pyqtgraph, when you execute your own script with pyqtgraph, it might throw error messages. There is a bug pyqtgraph with python 3.8. To resolve this, please follow below steps.

```
Traceback (most recent call last):
  File "c:\Users\hkchoi\Desktop\tmp\real.py", line 3, in <module>
    import pyqtgraph as pg
  File "C:\Python38\lib\site-packages\pyqtgraph\__init__.py", line 204, in
<module>
    from .graphicsItems.VTickGroup import *
  File "C:\Python38\lib\site-
packages\pyqtgraph\graphicsItems\VTickGroup.py", line 7, in <module>
    from .. import functions as fn
  File "C:\Python38\lib\site-packages\pyqtgraph\functions.py", line 17, in
<module>
    from . import debug
  File "C:\Python38\lib\site-packages\pyqtgraph\debug.py", line 11, in
<module>
    from . import ptime
  File "C:\Python38\lib\site-packages\pyqtgraph\ptime.py", line 24, in
<module>
    cstart = systime.clock()  ### Required to start the clock in windows
AttributeError: module 'time' has no attribute 'clock'
```

- **How to resolve**

```
1. Please install git first accordingly
   - https://git-scm.com/download/win
   - https://stackoverflow.com/questions/26620312/installing-git-in-path-with-github-client-for-
windows

2. After you installed Git window version, please add git Paths to the
environment.
```

```
3. After done, please restart your lab top.

4. In CMD, please check 'git' installed correctly.

5. Install pyqtgraph patch.
   - https://github.com/pyqtgraph/pyqtgraph/issues/1077
   - pip install git+https://github.com/pyqtgraph/pyqtgraph@develop
```



환경 변수

hkchoi에 대한 사용자 변수(U)

변수	값
CLSID	{C3829496-B47C-D1C1-2CF9-26CD4157F...
OneDrive	C:₩Users₩hkchoi₩OneDrive
Path	C:₩;C:₩Users₩hkchoi₩AppData₩Local₩Mi...
TEMP	C:₩Users₩hkchoi₩AppData₩Local₩Temp

환경 변수 편집

```
C:₩
C:₩Users₩hkchoi₩AppData₩Local₩Microsoft₩WindowsApps
C:₩Program Files (x86)₩IDM Computer Solutions₩UltraEdit₩
C:₩Program Files₩Microsoft VS Code₩bin
C:₩Users₩hkchoi₩AppData₩Local₩GitHubDesktop₩bin
%USERPROFILE%₩AppData₩Local₩Microsoft₩WindowsApps
C:₩Python38₩Scripts
C:₩Python38
C:₩Python38₩Lib₩site-packages
C:₩Python38₩studyrootm
C:₩Users₩hkchoi₩AppData₩Local₩Programs₩Microsoft VS Code₩...
C:₩Program Files₩Git₩bin₩git.exe;C:₩Program Files₩Git₩cmd
```

```
PS C:\Temp> git
usage: git [--version] [--help] [-C <path>] [-c <name>=<value>]
           [--exec-path[=<path>]] [--html-path] [--man-path] [--info-path]
           [-p | --paginate | -P | --no-pager] [--no-replace-objects] [--bare]
           [--git-dir=<path>] [--work-tree=<path>] [--namespace=<name>]
           <command> [<args>]

PS C:\Python38> pip install git+https://github.com/pyqtgraph/pyqtgraph@develop

PS C:\Temp> pip3 list
Package           Version
----------------- --------------------
pyqtgraph         0.11.0.dev0+g932b975 🡐=======
```

5. Contributor

* Rengaramalingam A (rengahcl@gmail.com)

Contributed to the feature of "Easy Lab Replication", which originated from his shell script "scriptit.sh"