UMSD CLI User Manual

1. Overview

CLI program emulates one telnet command line interface for UMSD API verification, there will be one command for each UMSD API, and it validates both chip and API functions. The program has been tested on Linux (Ubuntu 14.04 LTS 32/64bit) platform with the telnet client terminal “Putty”, through Marvell USB2SMI adapter or RMU setup.

This host application is used to verify the USMD APIs and depends on third party library “libusb” and “libpcap”. These third party libraries should be installed in the target system. You can use the following command for X86 Ubuntu system to install the libraries.

Note: If you are using embedded system, you should port the “libusb” and “libpcap” to the embedded system by yourself.

2. CLI application setup

2.1 Telnet server

1) Install libusb for SMI access, run the command on Ubuntu 14.04 for example.
   sudo apt-get install libusb-1.0
2) Install libpcap for RMU access, run the command on Ubuntu 14.04 for example.
   sudo apt-get install libpcap0.8-dev
3) Switch to UMSD root folder and run the command to source the global building variables.
   source setenv
4) Switch to host/linux folder and run command to build the test program.
   make
5) Start the telnet sever application.
   sudo ./MCLI

Choose SMI Interface for example:

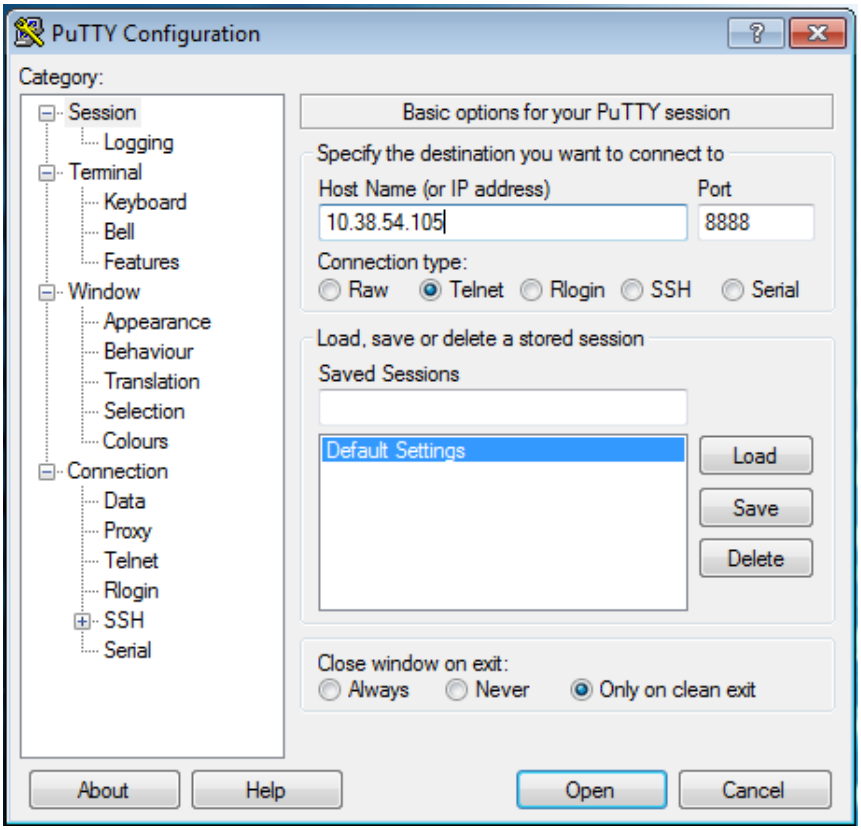
```
local3@local3-VirtualBox:~/unified_msd/host/linux$ sudo ./MCLI
[sudo] password for local3:
Interface List:
        0 : SMI Interface
        1 : SMI Multi-Chip
        2 : RMU Interface
Please select one interface: 0
Enter USB Port number(1 - 4):1
Only one USB Adapter connect(y/n)?y
Driver Load Success
Listening on port 8888 for telnet
```

Or choose the RMU DSA mode:

```
local3@local3-VirtualBox:~/zhougw/unified_msd/host/linux$ sudo ./MCLI
[sudo] password for local3:
Interface List:
        0 : SMI Interface
        1 : SMI Multi-Chip
        2 : RMU Interface
Please select one interface: 2
Select RMU Mode:
        0 : DSA Mode
        1 : Ether Type DSA Mode
Please select one RMU mode: 0
Please enter the Device number (hex, Dev=0 please enter 0x0 for example):0

================ NIC CARD List ================
(1) (null)
     eth0
(2) Linux netfilter log (NFLOG) interface
     nflog
(3) Linux netfilter queue (NFQUEUE) interface
     nfqueue
(4) Pseudo-device that captures on all interfaces
     any
(5) (null)
     lo
Please select NIC Index: 1

Using Interface: eth0

Driver Load Success
Listening on port 8888 for telnet
```

2.2 Telnet client Terminal

Using "*Putty*" as client terminal to connect to the server.



3. Commands usage

3.1 help usage

Using "*help*" to show all the commands description.

```
MCLI test environment

MCLI> help

Commands available:
  help                  Show available commands
  exit                  Exit from current program
  history               Show a list of previously run commands
  atu                   Show atu commands
  stu                   Show stu commands
  vlan                  Show vlan commands
  imp                   Show imp commands
  phy                   Show phy commands
  irl                   Show irl commands
  vct                   Show vct commands
  qav                   Show qav commands
  ptp                   Show ptp commands
```

Using "*cmd help*" to show all the sub command detail description

```
MCLI> atu help
atu help [subcmd]: Display atu command options
atu addEntry [options] : Creates the new entry in MAC address table.
atu delAll <deleteCMD> : delete all or all non-static atu entry
atu delAllInDB <deleteCMD> <fid>: delete all atu entry based on the DB number
atu move <moveCMD> <moveFrom> <moveTo>: move all or all non-static atu entry
atu moveInDB <moveCMD> <fid> <moveFrom> <moveTo>: move the atu entry based on fid
atu delEntry <macAddr> <fid> : delete the atu entry
atu getVio : get the atu Violation data
atu findEntry <macAddr> <fid> : find atu entry
atu setAging <timeout> : set the atu entry aging time out
atu getAging : get the atu entry aging time out
```

Using "*cmd help subcmd*" to show the parameter description and example

```
MCLI> atu help addEntry
atu addEntry [options]
        [options](default value for option item is 0):
        options for atu entry, format as: -macAddr aabbccddee00 -state 7
        -macAddr    : mac address, format as aabbccddee00
        -lag        : trunk member or LAG
        -portVec    : port vector
        -fid        : ATU MAC Address Database number
        -state      : atu entry state
        -FPri       : MAC frame priority data
        -QPri       : MAC Queue priority data
Example:
        atu addentry -macaddr 112233445566 -state 0xe -portvec 3 -fid 1
```

3.2 Arrow key handle

This CLI application can handle the following arrow keys: "*Tab*", "*Backspace*", "*left/right/up/down*".

"*Tab*": Tab completion, support the single completion and double tab for more than one completion.

```
MCLI> tcam addEntry 0 -frame
-frameType -frameTypeMask -frameOctet -frameOctetMask

MCLI> tcam addEntry 0 -frame
```

"*Backspace*": backspace character

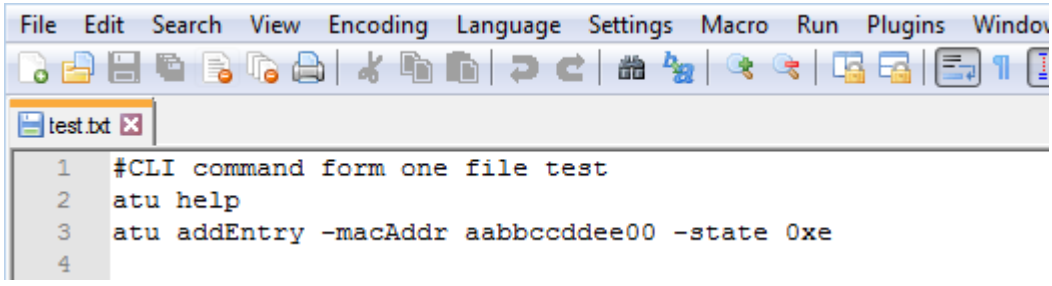
"*left/right*": left/right cursor motion

"*up/down*": show one history item

3.3 File operation

The commands can be invoked from file by type the command "file <filename>". Write the command in the file line by line and the prefix "#" means this line is comment.

Note: the file must be in server side, for example, the telnet sever is running on one Linux PC and the file is on the same PC with path: */home/netter/test/test.txt*




```
MCLI> file "/home/netter/test/test.txt"
atu addEntry [options] : Creates the new entry in MAC address table.
atu delAll <deleteCMD> : delete all or all non-static atu entry
atu delAllInDB <deleteCMD> <fid>: delete all atu entry based on the DB number
atu move <moveCMD> <moveFrom> <moveTo>: move all or all non-static atu entry
atu moveInDB <moveCMD> <fid> <moveFrom> <moveTo>: move the atu entry based on fid
atu delEntry <macAddr> <fid> : delete the atu entry
```