

# USIM Modifier

User Guide

Ming



2018

# Content

Content .....	1
History .....	2
Abstract .....	3
System Requirement .....	3
Special Thanks .....	3
System Features.....	3
Provided Plugins .....	3
Install “USIM Modifier” .....	4
Upgrade “USIM Modifier” .....	4
Install Requirement Packages .....	4
Environment Check.....	5
Start to using “USIM Modifier” .....	5
Plugins .....	6
atr .....	7
card_info.....	7
iccid.....	8
imsi .....	8
mccmnc .....	9
spn .....	9
msisdn .....	9
gid .....	10
pin_cache .....	10
send .....	11
dir .....	11
arr .....	11

# History

Revision	Common	Date
V1.0	Initial release	2018/12/15
V1.1	Install requirement packages on MAC OS X	2019/1/14
V1.2	Fixed some minor typo	1029/1/26

# Abstract

Sometimes we need to modify MCC/MNC, SPN or GID1 to verify some specific issue, so we didn't need to customized the “Test USIM” with powerful tool, that’s why to implement it.

## System Requirement

- Windows/Linux/MAC platforms
- Python 3.x (not compatible with Python 2)
- GIT
- PC/SC Smart Card Reader
- Requirement Packages of Python
  - [colorama](#)
  - [lxml](#)
  - [pyscard](#)

## Special Thanks

- Brian Beck: [switch class](#)

## System Features

- Command Line Interactive mode
- Auto detected locale for language and expandable.
- Python logging mechanism supported.
- Design by “Plugin” architecture and extendable.

## Provided Plugins

Information	<ul style="list-style-type: none"><li>➤ <b>card_info:</b> show the contents of ICCID, IMSI, MCC/MNC, SPN, GID1 &amp; GID2.</li><li>➤ <b>atr:</b> show the value of ATR (Answer To Reset).</li></ul>
-------------	---

<b>Customization</b>	<ul style="list-style-type: none"> <li>➤ <b>iccid</b>: display/modify the value of EF_ICCID.</li> <li>➤ <b>imsi</b>: display/modify the value of EF_IMSI.</li> <li>➤ <b>mccmnc</b>: display/modify the MCC/MNC (included MNC digits of EF_AD).</li> <li>➤ <b>gid</b>: display/modify the value of EF_GID1 &amp; EF_GID2.</li> <li>➤ <b>spn</b>: display/modify the value of EF_SPN.</li> <li>➤ <b>msisdn</b>: display/modify the value of EF_MSISDN.</li> </ul>
<b>Security</b>	<ul style="list-style-type: none"> <li>➤ <b>pin_cache</b>: store the PIN1/ADM code for verify automatically.</li> </ul>
<b>Expert</b>	<ul style="list-style-type: none"> <li>➤ <b>dir</b>: show all records of EF_DIR.</li> <li>➤ <b>arr</b>: show all records of EF_ARR (Under MF or ADF)</li> <li>➤ <b>send</b>: send the APDU command directly</li> </ul>

## Install “USIM Modifier”

git clone [https://github.com/minghsu/usim\\_modifier.git](https://github.com/minghsu/usim_modifier.git)

## Upgrade “USIM Modifier”

git pull

## Install Requirement Packages

<i>Platform</i>	<i>Steps</i>
<i>Linux</i>	<pre>linux@ubuntu:/\$ pip3 install colorama linux@ubuntu:/\$ sudo apt-get install swig linux@ubuntu:/\$ sudo apt-get install libpcsclite-dev linux@ubuntu:/\$ sudo pip3 install pycard</pre>
<i>Windows</i>	TBD
<i>Mac OS</i>	<p>Pre-condition: “HomeBrew” must be installed.</p> <pre>mac@osx:/\$ pip3 install colorama mac@osx:/\$ pip3 install lxml mac@osx:/\$ brew install swig (PS. <i>install “swig” by homebrew</i>) mac@osx:/\$ pip3 install pycard</pre>

# Environment Check

After installed all requirement packages, you can use “env\_check.py” to verify your system environment, if you get the same output with below picture, you can use “USIM Modifier” to customize your “Test USIM” card.

```
CHIH-MINGde-MBP:usim_modifier chih-minghsu$ ./env_check.py

USIM modifier environment checking ...

colorama package installed: Yes
lxml package installed: Yes
pyscard package installed: Yes

Result: All packages were installed
CHIH-MINGde-MBP:usim_modifier chih-minghsu$
```

## Start to using “USIM Modifier”

Please type “./usim\_modifier.py” (or type “python3 usim\_modifier.py”) on the command window and should get the similar message with below example.

```
CHIH-MINGde-MBP:usim_modifier chih-minghsu$ ./usim_modifier.py

USIM modifier Version 2.0 for Python 3, (C)2018 Hsu Chih-Ming

Connecting to "Alcor Micro AU9520" card reader ...
1) PIN1 verify
Please input PIN1 Code (4 ~ 8 digits, Decimal), press ENTER key to terminated: 1234
2) ADM verify
Please input ADM Key (16 digits, Hexadecimal), press ENTER key to skip:
3) Displayed the card info
ICCID: 89860009191190000108
IMSI: 466920123456789
MCC/MNC: 466/92
SPN: MING (16)
GID1: 88 FF FF FF FF FF FF (8)
GID2: 99 FF FF FF FF FF FF (8)
4) Security Information
PIN1 Enabled: True, PIN1 Verified: True, ADM Key Verified: False
5) Help Message
Type 'exit' to exit, 'plugin' for summary of supported plugins.
6) Ready for command
USIM modifier$
```

Let us to step by step to describe above example.

1. **PIN1 verify:** Will appear this step when the “Test USIM” enabled the PIN1

and we must input correct PIN1 to verify, most USIM fields were require PIN1 verified to read it.

2. **ADM verify:** Most USIM fields were requirement ADM verified to update content, we can skip it but will lost the “UPDATE” capability for “Test USIM”.
3. **Card Info:** Just auto executed the “card\_info” plugin.
4. **Security Info:** Show the “PIN1/ADM” information.
5. **Help Message:** Yes, just provide “exit” & “plugin” commands.
6. **Ready:** When you saw the “USIM modifier\$” indication, mean the “USIM modifier” was ready.

## Plugins

In this chapter, we will describe how to use the plugins, but you can type “plugin” command first to get list of supported plugins.

```
USIM modifier$ plugin
Plugin Name  Version  Summary
iccid 1.00 > Display or modify the value of ICCID.
spn 1.00 > Display or modify the value of SPN.
dir 1.00 > Displayed all contents of EF_DIR file.
card_info 1.00 > Displayed the current status of USIM.
send 1.00 > Send the APDU command to USIM directly
mccmnc 1.00 > Display or modify the value of MCC/MNC.
atr 1.00 > Displayed the value of Answer To Reset (ATR).
pin_cache 1.00 > Cache the PIN1/ADM code to xml file for future verify automatically.
arr 1.00 > Displayed all contents of EF_ARR file.
gid 1.00 > Display or modify the value of GID1/GID2.
msisdn 1.00 > Display or modify the value of MSISDN.
imsi 1.00 > Display or modify the value of IMSI.

Type '[plugin name] help' for more information about the plugin.
USIM modifier$
```

We can saw the “name”, “version” & “summary” of the plugin, and the bottom message indicated we can type “[plugin name] help” to get some help message of this plugin, below screenshot are using “mccmnc” & “iccid” plugins for example.

```

USIM modifier$ mccmnc help

Usage:
- mccmnc [mcc=xxx] [mnc=xxx]
Example:
- mccmnc
  > MCC/MNC: 466/92
- mccmnc mcc=320
  > MCC/MNC: 320/92
- mccmnc mnc=01
  > MCC/MNC: 466/01
- mccmnc mcc=001 mnc=01
  > MCC/MNC: 001/01

USIM modifier$ iccid help

Usage:
- iccid [set=iccid] [format=raw]

Example:
Original: 89860009191190000108
- iccid
  > ICCID: 89860009191190000108
- iccid format=raw
  > ICCID: 98 68 00 90 91 11 09 00 10 80
- iccid set=1234
  > ICCID: 12340009191190000108
- iccid set=12340009191190004321
  > ICCID: 12340009191190004321

PS. Suggest to verify ICCID with Luhn algorithm by https://planetcalc.com/2464/ first

```

## atr

It's a very simple plugin to display the ATR (Answer to Reset), that's a message output by a contact Smart Card conforming to [ISO/IEC 7816](https://www.iso.org/standards/std/7816.html) standards.

```

USIM modifier$ atr

ATR: 3B 9F 94 80 1F C7 80 31 E0 73 FE 21 13 57 86 85 03 86 98 42 18 AE

USIM modifier$ █

```

## card\_info

Displayed all contents of EF\_ICCID, EF\_IMSI, MCC/MNC, EF\_SPN, EF\_GID1 & EF\_GID2 fields.

```

USIM modifier$ card_info

ICCID: 89860009191190000108
IMSI: 466920123456789
MCC/MNC: 466/92
SPN: MING (16)
GID1: 88 FF FF FF FF FF FF FF (8)
GID2: 99 FF FF FF FF FF FF FF (8)

USIM modifier$ █

```



The “card\_info” plugin just called the “iccid”, “imsi”, “mccmnc”, “spn” & “gid” plugins and sort the return message.

## iccid

The “iccid” plugin can displayed the contents of EF\_ICCID with two format type, and update the content by partially or fully.

```
USIM modifier$ iccid help

Usage:
- iccid [set=iccid] [format=raw]

Example:
Original: 89860009191190000108
- iccid
> ICCID: 89860009191190000108 << Human Readable Format
- iccid format=raw
> ICCID: 98 68 00 90 91 11 09 00 10 80 << Raw Format
- iccid set=1234
> ICCID: 12340009191190000108 << Partially Update (4 digits)
- iccid set=12340009191190004321
> ICCID: 12340009191190004321 << Fully Update

PS. Suggest to verify ICCID with Luhn algorithm by https://planetcalc.com/2464/ first
```

If you want to modify the EF\_ICCID, please pay attend the **valid ICCID** is need to meet “**Luhn algorithm**”, you can pre-check with [planetcalc](https://planetcalc.com/2464/) website.

## imsi

The “imsi” plugin have the same operation with “iccid” plugin, the plugin will only update EF\_IMSI field and didn’t consider the length of MNC, if you need to update “MCC/MNC”, use “mccmnc” plugin to modify.

```
USIM modifier$ imsi help

Usage:
- imsi [set=imsi] [format=raw]

Example:
Original: 001010123456789
- imsi
> IMSI: 001010123456789
- imsi format=raw
> IMSI: 08 09 10 10 10 32 54 76 98
- imsi set=12345
> IMSI: 123450123456789
- imsi 466979876543210
> IMSI: 466979876543210
```

# mccmnc

The “mccmnc” plugin can modify the MCC/MNC values, and update correct length of MNC to EF\_AD field.

```
USIM modifier$ mccmnc help

Usage:
- mccmnc [mcc=xxx] [mnc=xxx]
Example:
- mccmnc
  > MCC/MNC: 466/92
- mccmnc mcc=320
  > MCC/MNC: 320/92
- mccmnc mnc=01
  > MCC/MNC: 466/01
- mccmnc mcc=001 mnc=01
  > MCC/MNC: 001/01
```

# spn

Show the current content and maximum length of EF\_SPN.

**Note: Only support ACSII coding, others are not support.**

```
USIM modifier$ spn help

Usage:
- spn [set=XXXXXX] [format=raw]

Example:
- spn
  > SPN: MAI TEST (16)
- spn format=raw
  > SPN: 01 4D 41 49 20 54 45 53 54 FF FF FF FF FF FF FF FF
- spn set=Orange
  > SPN: Orange
- spn set="My SIM"
  > SPN: My SIM
```

# msisdn

We can't easy find the “MSISDN” editor on current smart phone device, and we can use this plugin to modify.

To update EF\_MSISDN field didn't need “ADM” key verify, mean we can modify the operator's USIM card if “PIN1” verified.

```

USIM modifier$ msisdn help

Usage:
- msisdn [id=XX] [name=XXXXXX] [num=XXXXXX] [format=raw]

Example:
- msisdn
> EF_MSISDN #1 - Name: [Empty Content] (14), Number: 0928000000
> EF_MSISDN #2 - Name: [Empty Content] (14), Number: 0928000000
- msisdn format=raw
> EF_MSISDN #1 - FF FF FF FF FF FF FF FF FF FF FF FF FF FF 06 81 90 82 00 00 00 FF FF FF FF FF FF FF
> EF_MSISDN #2 - FF FF FF FF FF FF FF FF FF FF FF FF FF FF 06 81 90 82 00 00 00 FF FF FF FF FF FF FF
- msisdn id=1 name=Orange num=+886919001122
> EF_MSISDN #1 - Name: Orange (14), Number: +886919001122
> EF_MSISDN #2 - Name: [Empty Content] (14), Number: 0928000000
- msisdn id=2 name="My Test SIM"
> EF_MSISDN #1 - Name: Orange (14), Number: +886919001122
> EF_MSISDN #2 - Name: My Test SIM (14), Number: 0928000000

PS. For update MSISDN record, the "id" is a mandatory argument

USIM modifier$ █

```

## gid

Both GID1/GID2 can be read and update by this plugin.

```

USIM modifier$ gid help

Usage:
- gid [gid1=xxxxxxx] [gid2=xxxxxxx]

Example:
- gid
> GID1: FF FF FF FF FF FF FF FF
> GID2: FF FF FF FF FF FF FF FF
- gid gid1=12
> GID1: 12 FF FF FF FF FF FF FF
> GID2: FF FF FF FF FF FF FF FF
- gid gid2=1234567890ABCDEF
> GID1: FF FF FF FF FF FF FF FF
> GID2: 12 34 56 78 90 AB CD EF

USIM modifier$ █

```

## pin\_cache

The feature will store the "PIN1" & "ADM" key to a xml file, you can find the .xml file was stored under "cache\_file" folder, and file name will follow by ICCID.

After created the pin cache file, system will auto verify the "PIN1" & "ADM" key from next time.

**Note: Some "Test USIM" was configured same value of EF\_ICCID.**

```
USIM modifier$ pin_cache help

Usage:
  pin_cache pin1=xxxxxxx adm=xxxxxxxxxxxxxxxx
  - pin1: 4 ~ 8 digits
  - adm: 16 HEX digits

Example:
  pin_cache pin1=1234 adm=5555555555555555

PS. Using the ICCID as main file name.
```

## send

System will send the raw "APDU" command to "Test USIM" directly.

```
USIM modifier$ send help

Usage:
  - send XXXXXX

Example of 'SELECT MF':
  - send 00A40004023F00
  > , 61 1F

USIM modifier$
```

**dir**

Just show the contents of EF DIR field.

```
USIM modifier$ dir help

Usage:
- dir [format=raw]

Example:
- dir
  > #1 - AID: A0000000871002FF86FFFFF89FFFFFFFFF, Label: UniverSIM
- dir format=raw
  > #1 - AID: 61 1D 4F 10 A0 00 00 00 87 10 02 FF 86 FF FF 89 FF FF FF FF 50 09 55 6E 69 76 65 72 53 49 4D FF
```

**arr**

The “arr” plugin is for expert only, most people are didn’t to use this plugin, it will show all contents of “access rule reference”.

[illegible]