



# Grandstream Networks, Inc.

---

XML Configuration File Generator

User Guide

# XML Configuration File Generator User Guide

## Table of Content

<b>INTRODUCTION .....</b>	<b>3</b>
<b>FILES IN PACKAGE .....</b>	<b>4</b>
FILES DESCRIPTION .....	4
<b>USING XML CONFIGURATION FILE GENERATOR .....</b>	<b>5</b>
THE PROCESS OF USING XML CONFIGURATION FILE GENERATOR .....	5

---

**Table of Figures**  
**XML Configuration File Generator User Guide**

Figure 1: Filling information to the configuration file example..... 5

Figure 2: Commented P-values and valid P-values..... 6

Figure 3: The converted XML file with commented P-values and valid P-values ..... 6

Figure 4: Add necessary information into CSV file ..... 7

Figure 5: Another CSV file example..... 8

Figure 6: Execute the command to generate XML configuration files ..... 8

Figure 7: The XML file after the generating process..... 9

## INTRODUCTION

XML Configuration File Generator is a tool to generate XML configuration files, and the users can provision the XML configuration files to SIP phones to change its configurations. An XML configuration file consists of a list of name-value pairs. P-values are the elements in XML configuration files, and the values of the elements represent the values for that particular configurations in the SIP phones.

This user guide aims to help users to understand how to generate XML configuration files for a specific SIP phone under the Linux OS system.

## FILES IN PACKAGE

The package includes three files: XML Configuration File Generator, Configuration File Example (TXT file), and CSV File. Below are the descriptions for each file.

### FILES DESCRIPTION

<b>XML Configuration File Generator</b>	XML Configuration File Generator can convert Configuration Template TXT file into XML file under the Linux OS system.
<b>Configuration File Example (TXT File)</b>	An example of Configuration Template with some P-values which could be used for getting the XML file.
<b>CSV File</b>	A list includes MAC address, SIP User ID, Authentication ID, and Authentication Password, all of these will be added into the converted XML configuration file.

## USING XML CONFIGURATION FILE GENERATOR

### THE PROCESS OF USING XML CONFIGURATION FILE GENERATOR

1. Save XML Configuration File Generator, the configuration file example, and the CSV file in the same directory on Linux OS system.
2. Open configuration file example, fill the pre-provisioned P-values such as Account Name, SIP Server, SIP User ID, and SIP Authenticate Password into the TXT file. Moreover, users also can configure other more options manually before converting. As the figure shows below.

```
#####
## Example P values                                     ##
#####
# Account Active. 0 - No, 1 - Yes. Default value is 0
# Number: 0, 1
P271 = 1

# Account Name
P270 = Grandstream

# SIP Server
P47 = grandstream.com

# SIP User ID
P35 =

# SIP Authenticate ID
P36 =

# SIP Authenticate Password
P34 =

# Name (Display Name, e.g., John Doe)
P3 = Grandstream Network

#####
## Example P values End                               ##
#####
```

Figure 1: Filling information to the configuration file example

**Note:**

- a) To read User ID, Password, Auth ID in the MAC.csv, please add the corresponding P value of User ID in this TXT template to let the script know which the device is. The P value of User ID is different in different devices:
  - i. In DP715, GXW40xx, GXW42XX, HT704, the corresponding P values are P4060.

- ii. In GXW410X, the corresponding P values are P3060.
  - iii. In all the other devices, the corresponding P values are P35.
- b) XML Configuration File Generator will skip some characters in TXT file which are commented, and only converts the valid P values, the other characters will remain the same. For example, if the P values are commented, after converting, the P values will not be converted; if the P values are valid, it will be converted into XML file. As the figure shows below:

```
# Disable SIP NOTIFY Authentication. 0 -No, 1 - Yes. Default is No.
# Number: 0, 1
# P4428 = 0&d'd'd<d>
# P values added for Multicast Paging
# P1566 - P1588
-----
# P values added for Account 1
#-----
# Custom Call-info for Auto Answer.
# String
P2356 = d<d>d'd'd&
```

The characters in comment won't be converted.

The characters in valid P value will be converted.

Figure 2: Commented P-values and valid P-values

```
<!--# P4428 = 0&d'd'd<d>-->
<!--# P values added for Multicast Paging-->
<!--# P1566 - P1588-->
<!--#----->
<!--# P values added for Account 1-->
<!--#----->
<!--# Custom Call-info for Auto Answer. -->
<!--# String-->
<P2356>d< > d > ; d " ; d ' ; d & ; </P2356>
```

Figure 3: The converted XML file with commented P-values and valid P-values

3. Add the MAC address, SIP User ID, Authentication ID, and Authentication Password into the CSV file, these values will be added into the XML file after converting. As the figure shows below:

```

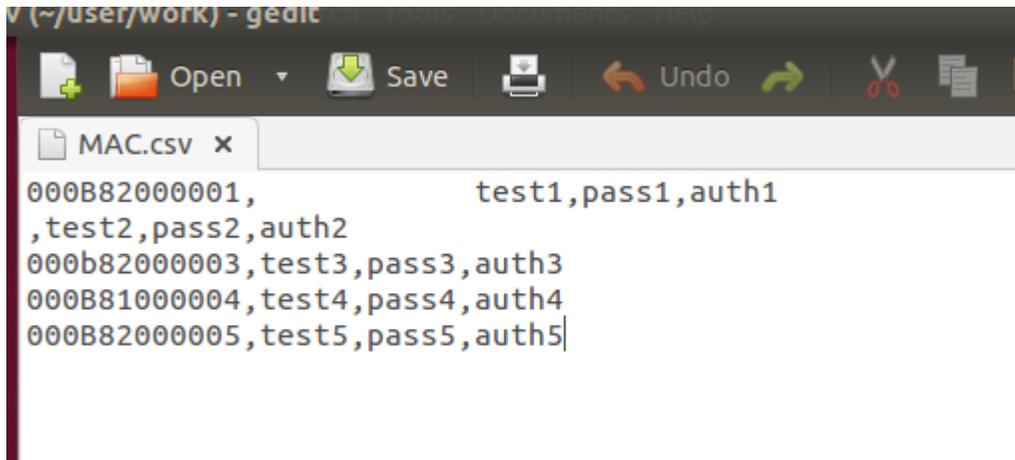
root@william-Satellite-L675D: /home/william/user/work
GNU nano 2.2.6 File: MAC.csv
000B82000001, test1, pass1, auth1
000B82000002, test2, pass2, auth2
000B82000003, test3, pass3, auth3
000B82000004, test4, pass4, auth4
000B82000005, test5, pass5, auth5

[ Read 5 lines (Converted from DOS format) ]
^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos
^X Exit ^J Justify ^W Where Is ^V Next Page ^U UnCut Text ^T To Spell

```

**Figure 4: Add necessary information into CSV file**

- a) The “Space” in MAC.csv file will be ignored by Configuration File Converter.
- b) If the MAC address in the MAC.csv file is not valid, the corresponding XML configuration file will not be generated.
- c) Users may use 000B82xxxxxx / 000b82xxxxxx as the MAC address in the MAC CSV file, all the MAC addresses without this specified starting string will be regarded as invalid. As the figure shows below:



**Figure 5: Another CSV file example**

The 1st, 3rd, and 5th MAC addresses with the configurations will be converted to XML configuration file. The 2nd and 4th MAC addresses will not be considered as valid MAC addresses, so the corresponding XML configuration file will not be generated.

4. Run the converting command to generate the XML configuration files, the command format is:

```
bash (converting tool name) (directory path) (config template) (MAC.csv)
```

**For example:**

```
bash GS_Config_TXT_to_XML_v1.5.sh /home/William/user/work/ Config_Example.txt MAC.csv
```

The figure below shows the command in Terminal:

```
root@william-Satellite-L675D:/home/william/user/work# bash GS_Config_TXT_to_XML_
v1.6.sh /home/william/user/work/ Config_Example.txt MAC.csv
Notice: The Mac "000B81000004" is invalid for not starting with "000B82"
, please check the CSV file
Thank you for using this program!
Your XML files will be generated at: /home/william/user/work/
root@william-Satellite-L675D:/home/william/user/work#
```

**Figure 6: Execute the command to generate XML configuration files**

5. Finally, the XML configuration files will be generated in the same directory with the P-values in the CSV file, users could use the generated XML files to provision a specified phone. As the figure shows below:

```

<?xml version="1.0" encoding="UTF-8" ?>
<!-- Grandstream XML Provisioning Configuration -->
<gs_provision version="1">
<mac>000B82000005</mac>
  <config version="1">
<!--#####
<!--###          Configuration Template Exmample for testing the converting tool
<!--#####
<!--##### -->
<!--## Example P values                                ##-->
<!--#####-->
<!--# Account Active. 0 - No, 1 - Yes. Default value is 0-->
<!--# Number: 0, 1-->
    <P271>1</P271>

<!--# Account Name-->
    <P270>Grandstream</P270>

<!--# SIP Server-->
    <P47>grandstream.com</P47>

<!--# SIP User ID-->
    <P35>test5</P35>

<!--# SIP Authenticate ID-->
    <P36>auth5</P36>

<!--# SIP Authenticate Password-->
    <P34>pass5</P34>

<!--# Name (Display Name, e.g., John Doe)-->
    <P3>GrandstreamNetwork</P3>

<!--##### -->
<!--## Example P values End                                ##-->
<!--##### -->

    </config>
  </gs_provision>

```

Figure 7: The XML file after the generating process