Developers Guide: Freedom Fone Sharicus 2.S.7/8

Contents

1 Overview

This guide should be read in conjunction with the Freedom Fone v2.S.x Installation and Developers Guide available from our website at www.freedomfone.org/page/documentation.

This Freedom Fone Sharicus v2.S.7 Installation and Developers Guide guide:

- Clarifies
  - Normalisation procedures and OfficeRoute settings
  - Management of Incoming SMS via OfficeRoute or USB dongle
  - Working with multiple OfficeRoute devices
  - Location of Log files
- Covers new features added in v2.S.7
  - Hardware detection and setup of USB devices managed by Gammu
  - The new SMS Centre menu option
  - Validation rules associated with SMS sending
  - Configuration of OfficeRoute & Freedom Fone for Outbound SMS
  - Sending/Receiving SMS using one or more OfficeRoute device
  - OfficeRoute and SIP authentication

2 Normalisation procedure

Phone numbers are represented in many different ways as they enter Freedom Fone. The differences depend on GSM implementation and country specific standards. For example, the phone number +468112233, can also be represented as 00468112233 (double zero plus country prefix), %2B468112233 (urldecoded plus sign plus country prefix), or simply 08112233 (without country prefix).

All normalization is managed by Freedom Fone, not the gateways it is using, or the dispatcher and spooler which manage incoming events.

2.1 OfficeRoute normalization

The OfficeRoute should always be configured with NO normalization rules in place. The backup file for OfficeRoute, provided by Freedom Fone is configured that way.

To verify that your OfficeRoute is correctly configured in regards to normalization rules, go to *Telephony services → GSM Routing → Normalization*.

The page should not show any normalization rules in place.

2.2 Freedom Fone normalization

The normalization takes place on all incoming phone numbers to Freedom Fone. Phone numbers enter Freedom Fone in the following models; Bin, Poll, Message (incoming SMS) and Cdr (call data records from incoming calls).

The normalization rules work as following:

Step 1: URLDECODE numbers (replace %2B for a plus sign (+)

Step 2: If the number starts with a plus sign (+), replace it with double zero (00).

Step 3: If number starts without plus sign (+) or without double zero (00), strip first zero (0) (if any) , and append double zero (00) and country prefix (taken from settings page).

3 Hardware detection and setup of Gammu devices

3.1 Gammu-smsd

gammu-smsd is called from rc.local.

3.1.1 Freedom Fone v2S7 built from repository

Parameter	Action	Command
start	will start designated IMSI	/etc/init.d/gammu-smsd start <IMSI>
stop	will stop designated IMSI	etc/init.d/gammu-smsd stop <IMSI>
status	will show the configured IMSIs	/etc/init.d/gammu-smsd show

3.1.2 Freedom Fone v2S7 installed from ISO file & v2S8

The init.d script is now modified to take the following parameters:

Parameter	Action	Command
start	will start ALL configured IMSIs	/etc/init.d/gammu-smsd start
stop	will stop ALL configured IMSIs	/etc/init.d/gammu-smsd stop
clean	will clean the configurations and make a .bak file of each of them	/etc/init.d/gammu-smsd clean
show	will show the configured IMSIs including inactive .bak files	/etc/init.d/gammu-smsd show

3.2 Dongle management

To avoid trouble with USB dongles, please follow these guidelines:

1. Shut down Freedom Fone (halt the server) before you unplug or plug in a new USB dongle.
2. After removing or inserting a dongle, run the discovery and setup scripts again, to re-configure the devices.
3. Make sure that the SIM card you are using is not protected with a PIN code.
4. Make sure that the SIM card you are using is topped up with money.
5. Make sure that the SIM card you are using is not a “data only” card, but can manage calls and

SMS as well.

6. Do not switch SIM cards between different physical dongles.

NOTE: If you want to reset all Freedom Fone references to telephony devices in the system (as displayed on *Active GSM channels* and *Service mapping* pages in the Freedom Fone user interface), see **3.3.11 Reinitialising telephony device detection settings**.

3.3 Gammu Step-by-step

3.3.1 Step 1: SIM preparation

Insert the SIM card into your Gammu device. Make sure that the SIM card is not protected with a PIN code. The SIM card needs to be active (make sure that you can send/receive phone calls or SMS to/from the SIM card) and have credit (at least if you want to send SMS).

3.3.2 Step 2: Connect dongle

Insert the Gammu dongle into an idle USB slot of your Freedom Fone server.

3.3.3 Step 3: Stop FreeSWITCH and the Dispatcher

When prompted for a password, use the sharicus password (default is *sharicus*)

```
$ sudo su
# /etc/init.d/freeswitch stop
# /etc/init.d/dispatcher_in stop
```

If you have run the *discovery* script previously, you will also have to stop gammu daemons currently running.

Freedom Fone v2S7 built from repository

To see what gammu devices are connected:

```
# /etc/init.d/gammu-smsd status
```

Which displays

```
# /etc/init.d/gammu-smsd start <IMSI>
You need to specify the IMSI channel to start/stop
IMSIs configured:
648040400727892
```

If there are IMSI numbers displayed, run the following command for each IMSI number – each one represents a connected USB GSM dongle.

```
# /etc/init.d/gammu-smsd stop <IMSI>
```

In our example:

```
# /etc/init.d/gammu-smsd stop <648040400727892>
```

Freedom Fone v2S7 installed from ISO file & v2S8

To see what gammu devices are connected:

```
# /etc/init.d/gammu-smsd show
```

Which displays

```
# The IMSIs configured for gammu are:
648040400727892
```

If the gammu_smsd 'clear' parameter has been used previously, you may see the following display:

```
# The IMSIs configured for gammu are:
648040400727892.bak
```

You need not be concerned about the backup configuration files referenced.

If there are IMSI numbers displayed (excluding .bak files), run the following command to stop all connected USB GSM dongles.

```
# /etc/init.d/gammu-smsd stop
```

3.3.4 Step 4: Discovery

Run the discovery script, that scans the Freedom Fone server for available Gammu devices. You need to run this script as root, so do not forget to use the sudo command.

```
sudo php /opt/freedomfone/hardware_wizard/discovery.php
```

If you have one single Gammu dongle inserted, you should see something like this on your screen:

```
Scanning ttyUSB0
Array
(
    [0] => Device               : /dev/ttyUSB0
    [1] => Manufacturer         : Huawei
    [2] => Model                : E1550 (E1550)
    [3] => Firmware             : 11.608.13.02.00
    [4] => IMEI                 : 358811039772494
    [5] => SIM IMSI             : 648040400727892
)
Scanning ttyUSB1
Array
(
    [0] => No response in specified timeout. Probably phone not connected.
)

Scanning ttyUSB2
Array
(
    [0] => Device               : /dev/ttyUSB2
    [1] => Manufacturer         : Huawei
    [2] => Model                : E1550 (E1550)
    [3] => Firmware             : 11.608.13.02.00
    [4] => IMEI                 : 358811039772494
    [5] => SIM IMSI             : 648040400727892
)
```

Each USB device detected will be presented with two entries, one for voice and one for data. In this case, they are mapped to USB0 and USB2. It is normal that one USB entry between the voice and data channels is empty. An empty USB entry will be prompted with the following message:

```
       "No response in specified timeout. Probably phone not connected."
```

The discovery script creates two configuration files:

1) gammu.rc

```
sharicus@sharicus-2S7dev: more /opt/freedomfone/hardware_wizard/tmp/gammu.rc

[gammu]
port = /dev/ttyUSB0
connection = at

[gammu1]
port = /dev/ttyUSB1
connection = at

[gammu2]
port = /dev/ttyUSB2
connection = at
```

2) hardware-discovery

```
sharicus@sharicus-2S7dev: more /opt/freedomfone/hardware_wizard/tmp/hardware-
discovery

648040400727892,358811039772494,Huawei,/dev/ttyUSB2,E1550 (E1550)
```

The meaning of each field in the hardware-discovery file is the following:

```
Field 1: IMSI
Field 2: IMEI
Field 3: Manufacturer
Field 4: Device (USB)
Field 5: Model
```

3.3.5 Step 5: Setup

To configure the Gammu dongle, run the setup script (as sudo).

```
      sudo php /opt/freedomfone/hardware_wizard/setup.php
```

For each USB dongle detected, you will be prompted with the following questions:

Question 1: Do you want to enable this device (Y/N)

Answer: NO
Action: Do nothing.
No more questions need to be answered.

Answer: YES
Action: Go to Question 2

Question 2: Will the device be used for SMS only

Answer: NO
Action: Configure device with Voice support (FreeSWITCH).
No more questions need to be answered.
Your next steps should be
i) go to 3.2.7 Step 7: Restart FreeSWITCH and Dispatcher
ii)go to Freedom Fone > Dashboard > Service mapping and select the voice menu to answer calls to the SIM in this device.

Answer: YES
Action: Go to Question 3

Question 3: Will the device send SMS

Answer: NO
Action: Configure device as Gammu device. Allow SMS receiving only.

Answer: YES
Action: Configure device as Gammu device. Allow both sending/receiving of SMS.

Once the setup wizard is completed, the following configuration files are either updated or created.

1) /opt/freedomfone/hardware_wizard/tmp/hardware-discovery

This file, which was created by the discovery script, is updated with field 6- 12.

```
Field 1: IMSI
Field 2: IMEI
Field 3: Manufacturer
Field 4: Device (USB)
Field 5: Model
Field 6: Enable (0 or 1)
Field 7: Gateway (freeswitch or gammu)
Field 8: Inbound (0 or 1)
Field 9: Outbound (0 or 1)
Field 10: Interface id
Field 11: Send (0 or 1)
Field 12: Receive (0 or 1)
```

In our case, the dongle is configured to send and receive SMSs. It will not answer calls:

```
648040400727892,358811039772494,Huawei,/dev/ttyUSB2,E1550
(E1550),1,gammu,1,1,GM1-648040400727892,1,1
```

```
2) /opt/freedomfone/gammu-smsd/conf/gammu-smsdrc-<IMSI>
```

For each enabled dongle, a gammu-smsdrc-<IMSI> file is created with the following format:

```
[gammu]

port = /dev/ttyUSB2
connection = at

[smsd]
service = sql
driver = native_mysql
debuglevel = 3
Logfile = /opt/freedomfone/log/gammu-GM1-648040400727892.log
user = gammu
password = thefone
pc = localhost
MaxRetries = 5
database = gammu
phoneid = GM1-648040400727892

Send = 1
Receive = 1
```

3.3.6 Step 6: Start Gammu

Freedom Fone v2S7 built from repository

For each enabled Gammu device, you need to run a script that starts a Gammu daemon. To list all enabled Gammu devices, run the following command.

```
/etc/init.d/gammu-smsd status
```

Which displays

```
/etc/init.d/gammu-smsd start <IMSI>
You need to specify the IMSI channel to start/stop
IMSIs configured:
648040400727892
```

To start the gammu daemon for a certain IMSI, do the following

```
/etc/init.d/gammu-smsd start <IMSI>
```

In our case:

```
/etc/init.d/gammu-smsd start 648040400727892
```

Freedom Fone v2S7 installed from ISO file & v2S8

Start all the gammu daemons as follows

```
/etc/init.d/gammu-smsd start
```

3.3.7 Step 7: Restart FreeSWITCH and Dispatcher

```
# /etc/init.d/freeswitch start
# /etc/init.d/dispatcher_in start
```

3.3.8 Step 8: Review your configuration

Navigate with a browser to your Freedom Fone installation, and go to *Dashboard → Service Mapping*. This view shows how your USB devices are configured. The table shows the following information:

Active: Yes or No.

Enabled: Yes or No.

Gateway: Gammu (SMS only) or FreeSWITCH (Voice services)

Inbound: Yes (Allow incoming SMS) or No (Do not allow incoming SMS)

Outbound: Yes (Allow outgoing SMS) or No (Do not allow outgoing SMS)

IMSI: Unique identifier of SIM card

Manufacturer: Modem manufacturer

Model: Modem model name

Hardware id: USB device name

Interface id: Freedom Fone unique id for configured device (<gateway><id>-<IMSI>)

Service: Freedom Fone service to map the device to. This option is only available if device is configured to allow voice calls (FreeSWITCH).

3.3.9 Step 9: Map device to service

If you have configured one or more USB devices for Voice services (FreeSWITCH), you must map them to a certain voice service (IVR or LAM). Under *Dashboard → Service Mapping*, you can select an IVR or LAM for each voice-configured USB device. Once you have selected the services of your choice, you must press the "Create Dialplan" button, and the FreeSWITCH dialplan will be updated with your choices.

3.3.10 Step 10: Test the USB device

Check the USB devices are displayed on the *Dashboard → Active GSM channels* page with *Last updated* set to the current time.

The system is now configured to use the USB device for calls and/or SMS. The best way to test that things work is to place phone calls to the service to which the device is mapped, and to create an SMS batch with the USB device as gateway.

NOTE: If things don't work as you expect, it may be that key services like FreeSWITCH, the Dispatcher or Gammu daemons haven't been restarted. The easiest thing may be to restart your server and run your tests. However be warned that restarting your server may require you to reinitialise the dongle's detection settings using step 3.3.11 below.

3.3.11 Reinitialising telephony device detection settings

To reset references to the USB dongles, follow these steps.

1) Stop FreeSWITCH and the Dispatcher

When prompted for a password, use the sharicus password (default is *sharicus*)

```
$ sudo su
# /etc/init.d/freeswitch stop
# /etc/init.d/dispatcher_in stop
```

2) Log into Gammu database

```
# mysql -u gammu gammu -p
```

password =thefone

3) Delete all data from phones table

```
# delete from phones;
# exit;
```

4) Log into Freedomfone database

```
# mysql -u freedomfone freedomfone -p
```

password =thefone

5) Delete all data from office_route and channels tables

```
# delete from office_route;
# delete from channels;
# exit;
```

6) Stop all gammu process

Freedom Fone v2S7 built from repository

To find out the IMSI of the running processes, run:

```
# /etc/init.d/gammu-smsd status
```

Run the following command for each device connected:

```
# /etc/init.d/gammu-smsd stop <IMSI>
```

Freedom Fone v2S7 installed from ISO file & v2S8

Stop all the gammu daemons as follows:

```
# /etc/init.d/gammu-smsd stop
```

7) Remove old Gammu config files

```
# rm /opt/freedomfone/hardware_wizard/tmp/hardware-discovery*
# /etc/init.d/gammu-smsd clean
```

8) Run discovery.php and setup.php scripts to configure USB dongles.

```
# php /opt/freedomfone/hardware_wizard/discovery.php
# php /opt/freedomfone/hardware_wizard/setup.php
```

9) Start Gammu processes

Freedom Fone v2S7 built from repository

For each enabled Gammu device, you need to run a script that starts a Gammu daemon. To list all enabled Gammu devices, run the following command.

```
/etc/init.d/gammu-smsd status
```

Which displays

```
/etc/init.d/gammu-smsd start <IMSI>
You need to specify the IMSI channel to start/stop
IMSIs configured:
648040400727892
```

To start the gammu daemon for a certain IMSI, do the following

```
/etc/init.d/gammu-smsd start <IMSI>
```

In our case:

```
/etc/init.d/gammu-smsd start 648040400727892
```

Freedom Fone v2S7 installed from ISO file & v2S8

Start all the gammu daemons as follows

```
/etc/init.d/gammu-smsd start
```

10) Start FreeSWITCH and the Dispatcher

```
# /etc/init.d/freeswitch start
# /etc/init.d/dispatcher_in start
```

11) If you have designated one or more dongles to answer voice calls, remember to assign them to either a LAM or other IVR using the *Dashboard → Service Mapping* page.

3.3.12 Troubleshooting dongle settings

Suggestions to follow if the dongles do not answer calls or receive SMSs as expected.

Notes

a. USB dongles are tricky to manage and /dev/ttyUSB allocations for them can be disturbed by adding/removing other USB devices like flashsticks. Be sure to check your installation is still working after adding/removing USB devices.

b. On some computers /dev/ttyUSB allocations change without apparent reason. We have seen this problem on older computers with what we believe is a faulty electrical supply to the usb bus which causes devices to be reset. This is an impossible scenario as it can happen at any time, causing your IVR & SMS receiving to fail. Track the situation by periodically logging into the terminal and listing the devices as follows:

```
$ sudo su
```

default password is sharicus

```
# ls -al /dev/ttyUSB*
```

Output will look similar to this for a single dongle:

```
root@sharicus-2S7-Ubuntu1404:/home/sharicus# ls -al /dev/ttyUSB*
crw-rw---- 1 root dialout 188, 0 Jul 22 12:32 /dev/ttyUSB0
crw-rw---- 1 root dialout 188, 1 Jul 22 12:32 /dev/ttyUSB1
crw-rw---- 1 root dialout 188, 3 Jul 22 12:32 /dev/ttyUSB2
root@sharicus-2S7-Ubuntu1404:/home/sharicus#
```

If these allocations later change to something like the following, your service will stop answering and the device(s) will disappear from the GSMOpen section on the Dashboard Active GSM channels page.

```
root@sharicus-2S7-Ubuntu1404:/home/sharicus# ls -al /dev/ttyUSB*
crw-rw---- 1 root dialout 188, 0 Jul 22 12:44 /dev/ttyUSB0
crw-rw---- 1 root dialout 188, 1 Jul 22 12:44 /dev/ttyUSB1
crw-rw---- 1 root dialout 188, 3 Jul 22 12:44 /dev/ttyUSB3
root@sharicus-2S7-Ubuntu1404:/home/sharicus#
```

You can trying running through the setup again, but chances are you will not be able to continue to use the current computer with Freedom Fone versions running on Ubuntu 1404LTS. You could try running FFv2S7 which runs on Ubuntu 1204LTS, but we have experienced similar issues with some computers on that platform in the past.

Other options

1) Shut the server down, reseat the dongles and boot up again. Run through *Reinitialising telephony device detection settings* above again.

2) For dongles configured for voice and SMS

Check that the /dev/ttyUSB allocations for your dongle(s) match your /opt/freeswitch/conf/autoload_configs/gsmopen.conf.xml configuration file.

```
$ sudo su
```

default password is sharicus

```
# ls -al /dev/ttyUSB*
root@sharicus-2S7-Ubuntu1404:/home/sharicus# ls -al /dev/ttyUSB*
crw-rw---- 1 root dialout 188, 0 Jul 22 12:44 /dev/ttyUSB0
crw-rw---- 1 root dialout 188, 1 Jul 22 12:44 /dev/ttyUSB1
crw-rw---- 1 root dialout 188, 3 Jul 22 12:44 /dev/ttyUSB3
root@sharicus-2S7-Ubuntu1404:/home/sharicus#
```

Make a note of the two highest numbered ttyUSB values. In the example above they are 1 and 3.

```
gedit /opt/freeswitch/conf/autoload_configs/gsmopen.conf.xml
```

Check this snippet:

```
 <per_interface_settings>
  <interface id="1" name="FS1-648040602971361">
   <param name="controldevice_name" value="/dev/ttyUSB3"/>
   <param name="controldevice_audio_name" value="/dev/ttyUSB2"/>
   <param name="destination" value="5000"/>
  </interface>
 </per_interface_settings>
```

The *controldevice_name* parameter should be set to the highest ttyUSB value for your dongle.

The *controldevice_audio_name* parameter should be set to the second highest ttyUSB value for your dongle.

If there is a mismatch – as in the example above (the *controldevice_audio_name* parameter should be = /dev/ttyUSB1 for this example) – follow these steps:

a. Close /opt/freeswitch/conf/autoload_configs/gsmopen.conf.xml without making any changes

b. Stop freeswitch & dispatcher_in

```
# /etc/init.d/freeswitch stop
# /etc/init.d/dispatcher_in stop
```

c. Change settings in /opt/freeswitch/conf/autoload_configs/gsmopen.conf.xml

```
gedit /opt/freeswitch/conf/autoload_configs/gsmopen.conf.xml
```

d. Start freeswitch & dispatcher_in

```
# /etc/init.d/freeswitch start
# /etc/init.d/dispatcher_in start
```

e. Check to see if the USB device now appears on the *Dashboard → Active GSM channels* page with *Last updated* set to the current time. Refresh the page a few times over the space of 1-2 mins if necessary.

3) For SMS sending/receiving dongles (no voice)

If the dongle is correctly configured but doesn't automatically start when you reboot the computer, start it manually. You will notice that the Updated date in the Gammu table on the *Dashboard → Active GSM channels* page is not current when you visit or refresh this page.

```
$ sudo su
```

default password is sharicus

Start gammu processes for configured dongles:

Freedom Fone v2S7 built from repository

For each enabled Gammu device, you need to run a script that starts a Gammu daemon. To list all enabled Gammu devices, run the following command.

```
/etc/init.d/gammu-smsd status
```

Which displays

```
/etc/init.d/gammu-smsd start <IMSI>
You need to specify the IMSI channel to start/stop
IMSIs configured:
648040400727892
```

To start the gammu daemon for a certain IMSI, do the following

```
/etc/init.d/gammu-smsd start <IMSI>
```

In our case:

```
/etc/init.d/gammu-smsd start 648040400727892
```

Freedom Fone v2S7 installed from ISO file & v2S8

Start all the gammu daemons as follows

```
/etc/init.d/gammu-smsd start
```

Check if the Updated date in the Gammu table on the *Dashboard → Active GSM channels* is now current.

4) Check the Freedom Fone website Forum for other tips or information http://freedomfone.org/forum#/categories/hardware

4 SMS Centre

4.1 New database tables

The following three tables have been added to create the SMS Centre. They all belong to the database "freedomfone".

SMS_GATEWAYS

Describes an SMS gateway created through the GUI (*SMS Centre → SMS Gateways → Create new*).

Name	**Type**	**Description**
id	int(11)	Primary key
name	varchar(50)	Unique key
url	varchar(100)	URL to SMS gateway
username	varchar(50)	Credential
password	varchar(50)	Credential
api_key	varchar(100)	Credential
created	int(11)	System created data
modified	int(11)	System created data
comment	varchar(100)	Optional comment
gateway_code	varchar(2)	Two character code for gateway. Default 'CT' (Clickatell)

BATCHES

Describes an SMS batch created through the GUI (*SMS Centre → Create Batch*).

Name	**Type**	**Description**
id	int(11)	Primary key, unique id
body	varchar(200)	SMS body
name	varchar(50)	Unique key. Name of batch for GUI display.
filename	varchar(50)	Name of uploaded file with receivers' numbers.
created	int(11)	System created data
modified	int(11)	System created data
gateway_type	varchar(6)	IP_GW (Clickatell) or GSM_GW (OfficeRoute or Gammu)
status	tinyint(4)	1 (Success), 0 (Failure)
gateway_code	varchar(2)	CT (Clickatell) GM (Gammu) OR (OfficeRoute) PC (Panacea, for future use)
sms_gateway_id	int(11)	id of sms_gateway in use (NULL for GSM gateways)
sender_number	varchar(25)	Sender number (if provided)

SMS RECEIVERS

Manages receivers of an SMS batch with name and status.

Name	**Type**	**Description**
id	int(11)	Primary key, unique id
batch_id	int(11)	ID of batch
receiver	varchar(50)	Phone number of receiver
sms_gateway_id	int(11)	0 (Gammu)
apimsgid	varchar(35)	Session id provided by SMS gateway (Clickatell) Null for Gammu and OfficeRoute.
status	varchar(20)	Status code provided by SMS gateway.
gateway_id	int(11)	Only for Gammu. Id of corresponding entry in Gammu table “sentitems”. Used to keep track of status.

4.2 Status messages

There are two types of status involved in SMS Centre: Batch status and an individual SMS status per receiver.

4.2.1 Batch status

The batch status has a value of either True or False, and the criteria varies for each gateway type.

OfficeRoute: Outgoing SMS via OfficeRoute is managed via email. A mail is sent to a POP account in OfficeRoute, which is later processed by the device and purged. There is no way for Freedom Fone to know whether the sms has arrived at the receiver. Freedom Fone can only know if a mail was successfully sent to the OfficeRoute.

True if mail has successfully been sent to OfficeRoute.

Gammu: True if entry has been entered into the Gammu database.

Clickatell: Freedom Fone authenticates with Clickatell when a new batch is created. If the authentication is successful, the status value will be set to True.

4.2.2 Individual SMS status

OfficeRoute: Not available (empty field)

Gammu: Status of outgoing SMS via Gammu is stored in the table “sentitems” in the Gammu database, fetched by Freedom Fone and stored in the table “sms_receivers”, field “status”.

There are three different status codes

1. Gammu is not running: A Gammu device is detected and setup, but no Gammu daemon is running. In this case, the SMS will be stuck in the Gammu outbox table.

2. Sending OK: Message has successfully been sent.

3. Sending failed: Failure due to incorrect SMS receiver number, blocked SIM card, no credit or other reason related to the Gammu device.

Clickatell: The mapping between Clickatell status codes for individual SMS, and human readable strings is done in config.php. See table below for the complete mapping.

Status code	Message
001	Message unknown. The message ID is incorrect or reporting is delayed
002	Message queued. The message could not be delivered and has been queued for attempted redelivery
003	Delivered to gateway. Delivered to the upstream gateway or network (delivered to

	the recipient)
004	Received by recipient. Confirmation of receipt on the handset of the recipient
005	Error with message. There was an error with the message, probably caused by the content of the message itself
006	User canceled message delivery. The message was terminated by a user (stop message command) or by our staff
007	Error delivering message. An error occurred delivering the message to the handset
008	OK. Message received by gateway
009	Routing error. An error occurred while attempting to route the message
010	Message expired. Message has expired before we were able to deliver it to the upstream gateway. No charge applies.
011	Message queued for later delivery' , 'Message has been queued at the gateway for delivery at a later time (delayed delivery).
012	Out of credit. The message cannot be delivered due to a lack of funds in your account. Please re-purchase credits.
014	Maximum MT limit exceeded. The allowable amount for MT messaging has been exceeded.

4.3 Interfacing with Clickatell

Clickatell accepts phone numbers with the following format: country code + phone number. The country prefix should not include double zero (00) or plus sign (+), only the country code (1-3 digits) of the specific country.

Freedom Fone takes care of the normalization of the phone numbers when using Clickatell as the gateway. Freedom Fone allows the following three formats to be used:

- +<country prefix><phone number without leading zero>
- 00<country prefix><phone number without leading zero>
- <phone number>

Important note: Clickatell limits the number of phone numbers in a single batch to 100 due to its sending mechanism (HTTP). There is no such maximum limit for OfficeRoute and Gammu.

5 Validation rules applied

5.1 SMS gateways

Table field	Validation
name	Minimum 5 characters, unique string
username	Required
password	Required
url	Required
api_key	Numeric, required

5.2 Batches

Table field	Validation
body	Maximum 160 characters, required
filename	Required
name	Minimum 5 characters, required
sms_gateway_id	Required

5.3 SMS Receivers

No validation rules are applied by CakePHP, other than what is explained under "Interfacing with Clickatell".

6 Incoming SMS

Incoming SMS can come either via OfficeRoute or via a Gammu device (USB dongle).

6.1 OfficeRoute

Incoming SMSs to OfficeRoute is stored in a POP3 account inside the OfficeRoute. The POP3 daemon (/opt/freedomfone/office_route/pop3_daemon.php) is run by a cronjob every minute. The POP3 daemon retrieves incoming SMS from the POP3 account, and creates a FreeSWITCH event with the following script:

```
/opt/freeswitch/scripts/freedomfone/sms/officeroute_main.js
```

All events are logged in the following file:

```
/opt/freedomfone/log/pop3_daemon.log
```

The FreeSWITCH event is picked up by the dispatcher, and is stored in the spooler database. Freedom Fone (by means of a cronjob) is fetching spooler data every minute.

6.1.1 Troubleshooting Incoming SMS to OfficeRoute sims

Check the OfficerouteParam setting in /opt/freedomfone/office_route/pop3_daemon.php.

```
$ sudo su
```

default password is sharicus

```
# gedit /opt/freedomfone/office_route/pop3_daemon.php
```

If you're using a single OfficeRoute, make sure the file is saved with the following:

```
$_OfficerouteParam = $_OfficerouteParamSingle;   //Single Channel
//$_OfficerouteParam = $_OfficerouteParamMulti;   //Multi
```

If you're using multiple OfficeRoutes, make sure the file is saved with the following:

```
//$_OfficerouteParam = $_OfficerouteParamSingle;   //Single Channel
$_OfficerouteParam = $_OfficerouteParamMulti;   //Multi
```

If you've made any changes, make sure to stop and restart FreeSWITCH and the dispatcher:

```
# /etc/init.d/freeswitch stop
# /etc/init.d/dispatcher_in stop
# /etc/init.d/freeswitch start
# /etc/init.d/dispatcher_in start
```

Resend messages to all sims in the OfficeRoute(s) and confirm they are being received.

6.2 Gammu

Incoming SMSs to a Gammu device enter into the Gammu database and are stored in the “inbox” table.

The Gammu daemon (/opt/freedomfone/gammu-smsd/daemon/gammu_daemon.php) is run by a cronjob every minute. The Gammu daemon retrieves incoming SMS from the MySQL table “inbox”, and creates a FreeSWITCH event with the following script:

```
    /opt/freeswitch/scripts/freedomfone/sms/gammu_main.js
```

All events are logged into the following file:

```
    /opt/freedomfone/log/gammu_daemon.log
```

The FreeSWITCH event is picked up by the dispatcher, and is stored in the spooler database. Freedom Fone (by means of a cronjob) is fetching spooler data every minute.

7 Configuration of OfficeRoute & Freedom Fone for Outbound SMS

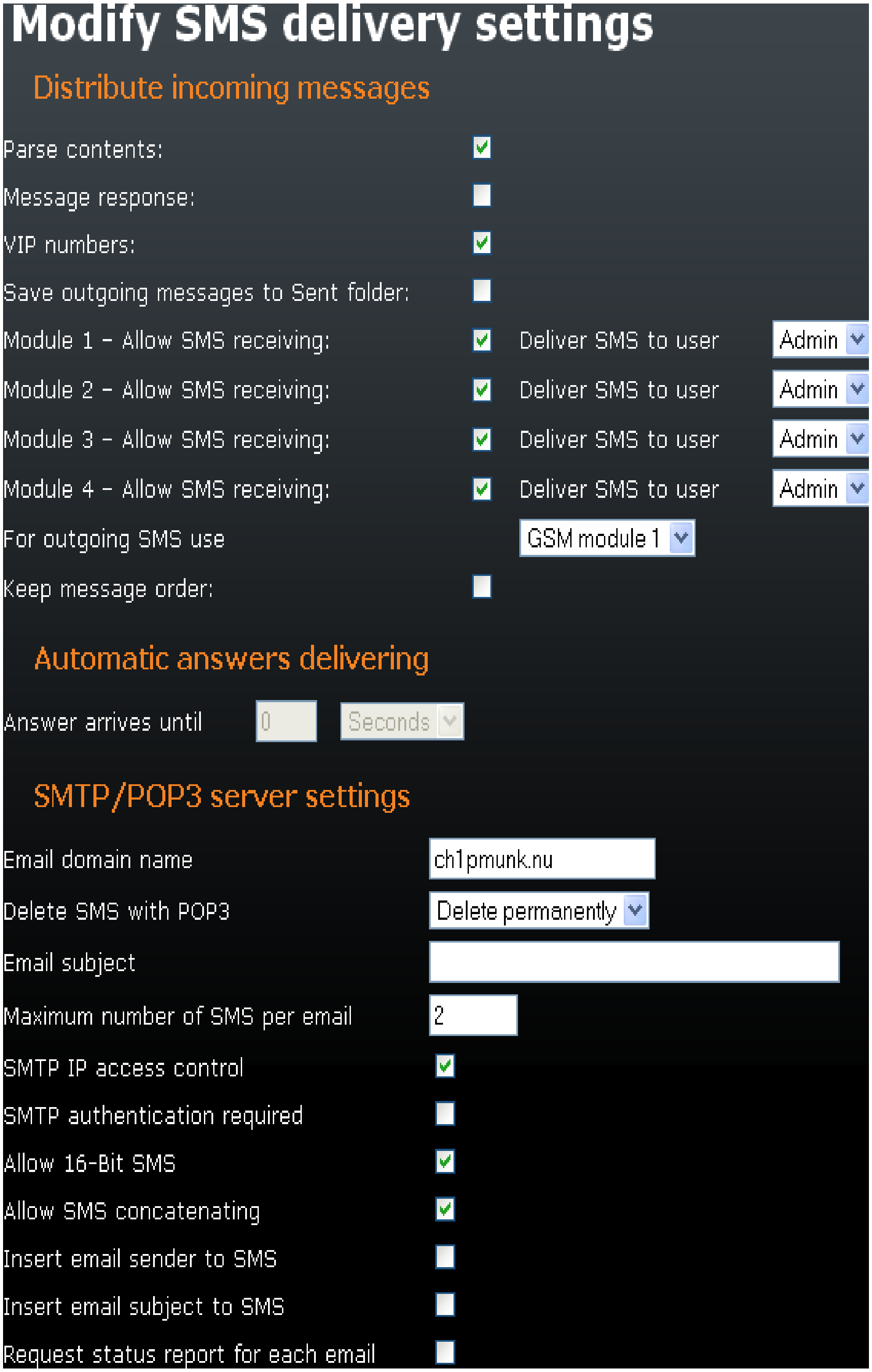
7.1 OfficeRoute

7.1.1 SMS delivery

Log in to OfficeRoute and configure the SMS Delivery settings (*Messaging → SMS delivery*).

Pay attention to the following settings:

- Message response: false
- Save outgoing messages to Sent folder: false
- For outgoing SMS use: select appropriate GSM module
- Email domain name: see instructions below
- Maximum number of SMS per email: 2
- SMTP IP access control: true
- SMTP authentication required: false
- Insert email sender to SMS: false
- Insert email subject to SMS: false



Modify SMS delivery settings

Distribute incoming messages

Parse contents:	☑		
Message response:	□		
VIP numbers:	☑		
Save outgoing messages to Sent folder:	□		
Module 1 - Allow SMS receiving:	☑	Deliver SMS to user	Admin
Module 2 - Allow SMS receiving:	☑	Deliver SMS to user	Admin
Module 3 - Allow SMS receiving:	☑	Deliver SMS to user	Admin
Module 4 - Allow SMS receiving:	☑	Deliver SMS to user	Admin
For outgoing SMS use		GSM module 1	
Keep message order:	□		

Automatic answers delivering

Answer arrives until 0 Seconds

SMTP/POP3 server settings

Email domain name	ch1pmunk.nu
Delete SMS with POP3	Delete permanently
Email subject	
Maximum number of SMS per email	2
SMTP IP access control	☑
SMTP authentication required	□
Allow 16-Bit SMS	☑
Allow SMS concatenating	☑
Insert email sender to SMS	□
Insert email subject to SMS	□
Request status report for each email	□

7.1.2 Email domain name

OfficeRoute needs an Email domain name for outgoing SMS due to its use of *postfix*.

Come up with a domain name that is not in use (and not likely to every be in use). To verify that the domain is not in use, run "*whois*" and see that no matching record is found.

```
root@fred:~# whois ch1pmunk.nu
------------------------------------------------------------------------
.NU Domain Ltd Whois service
NO MATCH for domain "ch1pmunk.nu" (ASCII):
Visit http://www.nunames.nu to determine if this domain has been
registered within the last 48 hours.
Owner and Administrative Contact information for domains
registered in .nu is available upon request from support@nic.nu
Copyright by .NU Domain Ltd - http://www.nunames.nu
------------------------------------------------------------------------
Database last updated: 2012-Nov-14 08:44:27 GMT
```

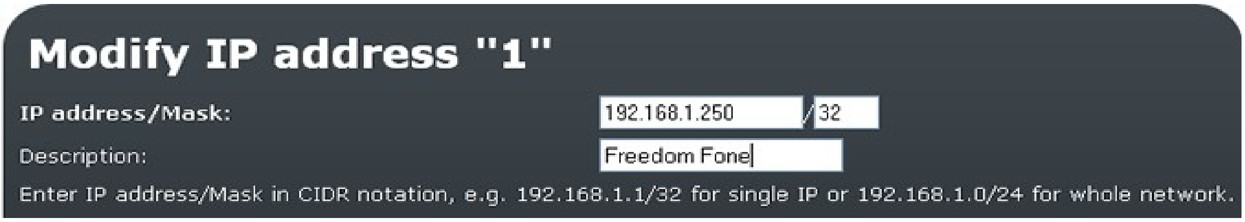
The default email domain setting is *ch1pmunk.nu*. If you decide to use something different, make sure to edit the OR_SNMP parameter in /opt/freedomfone/gui/app/Config/config.php

```
$config['OR_SNMP']= array(
                        array('ip_addr'   => '192.168.1.46',
                        'domain'      => 'ch1pmunk.nu',
                              'community' => 'public' ,
                              'object_id' => '1.3.6.1.4.1.6530.4.2.2.1'
                        ),
                );
```

7.1.3 SMTP IP access control

In order to allow only our Freedom Fone server to use OfficeRoute for outgoing SMS, we add an IP based access control. Add an entry with the IP address of your Freedom Fone server as following:

Messaging → SMS delivery → SMTP IP access → Add IP address



Modify IP address "1"

IP address/Mask: 192.168.1.250 / 32

Description: Freedom Fone

Enter IP address/Mask in CIDR notation, e.g. 192.168.1.1/32 for single IP or 192.168.1.0/24 for whole network.

If you plan to run Freedom Fone in a network with IP 192.168.1.x the filter should be 192.168.1.0/24

If the Freedom Fone server runs in another IP, use 192.168.1.250/24

The overall assumption is that Freedom Fone is not operating in a network with IP 192.168.1.x as there can be collisions with the OfficeRoute on IP 192.168.1.46 and Freedom Fone secondary IP address set to 192.168.1.250.

7.2 Freedom Fone

SMS are sent as emails to the OfficeRoute. To ensure that SMS can be queued, a local mail agent will receive the mails, to be relayed to OfficeRoute at a later stage.

Freedom Fone uses “Postfix” as mail agent and it needs to be configured on the Freedom Fone server to allow outgoing mail to OfficeRoute.

Two files needs to be edited:

- /etc/postfix/main.cf AND
- /etc/postfix/transport

In main.cf, add an entry for “transport_maps”. This line tells postfix that mails with certain domain name need to be dequeue via certain IP address.

```
# See /usr/share/postfix/main.cf.dist for a commented, more complete version

# Debian specific: Specifying a file name will cause the first
# line of that file to be used as the name. The Debian default
# is /etc/mailname.
#myorigin = /etc/mailname
transport_maps = hash:/etc/postfix/transport
myhostname = localhost
smtpd_banner = $myhostname ESMTP $mail_name (Ubuntu)
biff = no

# appending .domain is the MUA's job.
append_dot_mydomain = no

# Uncomment the next line to generate "delayed mail" warnings
#delay_warning_time = 4h

readme_directory = no

# TLS parameters
smtpd_tls_cert_file=/etc/ssl/certs/ssl-cert-snakeoil.pem
smtpd_tls_key_file=/etc/ssl/private/ssl-cert-snakeoil.key
smtpd_use_tls=yes
smtpd_tls_session_cache_database = btree:${data_directory}/smtpd_scache
smtp_tls_session_cache_database = btree:${data_directory}/smtp_scache

# See /usr/share/doc/postfix/TLS_README.gz in the postfix-doc package for
# information on enabling SSL in the smtp client.

alias_maps = hash:/etc/aliases
alias_database = hash:/etc/aliases
mydestination = localhost.localdomain, , localhost
```

```
relayhost =
mynetworks = 127.0.0.0/8 [::ffff:127.0.0.0]/104 [::1]/128
mailbox_size_limit = 0
recipient_delimiter = +
inet_interfaces = all
```

Create the file “transport”, and add the following line:

```
<email_domain_name> smtp:[<office_route_ip_address>]
```

In this example, the “transport” file looks as following:

```
ch1pmunk.nu smtp:[192.168.1.46]
```

Once you have created the file transport, run the command:

```
postmap /etc/postfix/transport
```

and restart postfix

```
/etc/init.d/postfix restart
```

The postfix log is located in /var/log/mail.log

7.2.1 Troubleshooting Outbound SMS via the OfficeRoute

Check postfix log entries in /var/log/mail.log. If you see the kind of error highlighted below, try setting SMTP IP access to 192.168.1.0/24.

```
Aug 28 15:54:11 ubuntu postfix/pickup[1581]: 021521A50B7: uid=33 from=<www-data>
Aug 28 15:54:11 ubuntu postfix/cleanup[8833]: 021521A50B7: message-
id=<20140828135411.021521A50B7@sharicus>
Aug 28 15:54:11 ubuntu postfix/qmgr[1582]: 021521A50B7: from=<www-data@sharicus>,
size=412, nrcpt=1 (queue active)
Aug 28 15:54:11 ubuntu postfix/pickup[1581]: 0A9511A50B6: uid=33 from=<www-data>
Aug 28 15:54:11 ubuntu postfix/cleanup[8833]: 0A9511A50B6: message-
id=<20140828135411.0A9511A50B6@sharicus>
Aug 28 15:54:11 ubuntu postfix/qmgr[1582]: 0A9511A50B6: from=<www-data@sharicus>,
size=412, nrcpt=1 (queue active)
Aug 28 15:54:11 ubuntu postfix/smtp[8838]: 021521A50B7:
to=<0772232680@ch1pmunk.nu>, relay=192.168.1.46[192.168.1.46]:25, delay=0.29,
delays=0.09/0.08/0.01/0.12, dsn=5.0.0, status=bounced (host
192.168.1.46[192.168.1.46] said: 550 Requested action not taken: IP check failed
(in reply to RCPT TO command))
```

8 Connecting multiple OfficeRoute devices

To use more than one OfficeRoute device, you need to add the device to the config.php file of Freedom Fone.

```
/opt/freedomfone/gui/app/Config/config.php

$config['OR_SNMP']= array(
                        array('ip_addr'     => '<ip_address_1>',
                                'domain'      => '<domain_1>',
                              'community'   => 'public' ,
                              'object_id'   => '1.3.6.1.4.1.6530.4.2.2.1'
                        ),
                        array('ip_addr'     => '<ip_address_2>',
                                'domain'      => '<domain_2>',
                              'community'   => 'public' ,
                              'object_id'   => '1.3.6.1.4.1.6530.4.2.2.1'
                        ),
                );
```

Next we need to instruct the POP daemon that connects to OfficeRoute and fetches mail from its POP accounts.

Open the following file:

```
/opt/freedomfone/office_route/pop3_daemon.php
```

Comment out the option that is not in use.

```
$_OfficerouteParam = $_OfficerouteParamSingle;   //Single Channel
$_OfficerouteParam = $_OfficerouteParamMulti;   //Multi
```

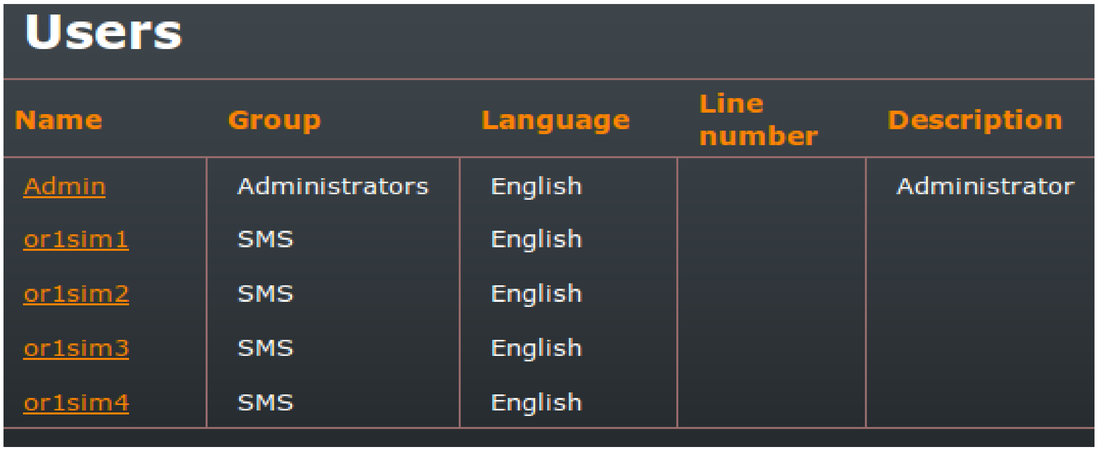
9 Sending/Receiving SMS using one or more OfficeRoute device

If you plan to use more than one OfficeRoute device **and/or** wish to distribute inbound SMSs into a different Inbox for each SIM card, you should load your OfficeRoute device with the following backup file:

https://dev.freedomfone.org/export/2298/trunk/office_route/backups/backup_officeroute_2.3.22_multiSMS_outSMS.tgz

The file will create four users, one for each SIM card (or1sim1, or1sim2, or1sim3 and or1sim4).

You can find the accounts under *User Management → Users*



Users

Name	Group	Language	Line number	Description
Admin	Administrators	English		Administrator
or1sim1	SMS	English		
or1sim2	SMS	English		
or1sim3	SMS	English		
or1sim4	SMS	English		

The password is “2n” (OfficeRoute default password) for all accounts. If you decide to change password (which is highly recommended), you need to reflect the changes in the configuration file of the pop3_daemon.php, which fetches incoming SMS from the pop accounts of the OfficeRoute.

Open the file /opt/freedomfone/office_route/config.php

Edit the “pass” field of either $_OfficerouteParamSingle (if you use one single OfficeRoute) or $_OfficerouteParamMulti (if you use more than one OfficeRoute).

9.1 Distribution of inbound SMS

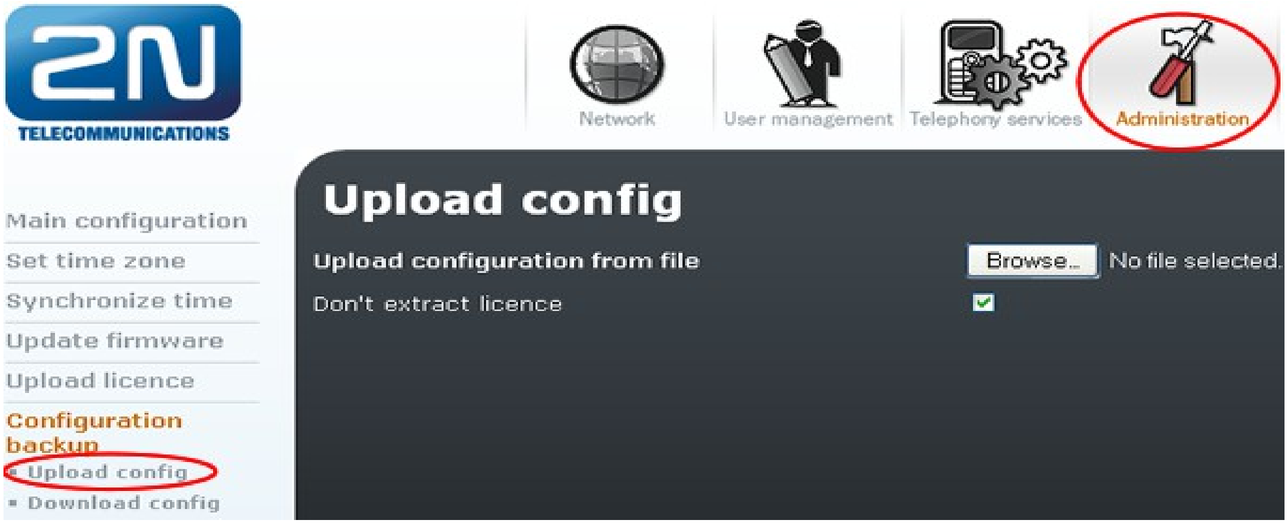
You can find SMS delivery settings under *Messaging → SMS Delivery.*

a) If you want SMSs received into **different** User inboxes by SIM card, use this configuration backup file: https://dev.freedomfone.org/export/2298/trunk/office_route/backups/backup_officeroute_2.3.22_multiSMS_outSMS.tgz
A copy should also be available in a folder on your Freedom Fone server desktop.

b) If you want all inbound SMSs to aggregate in **one common inbox** (Admin) use this configuration backup file: https://dev.freedomfone.org/browser/trunk/office_route/backups/backup_officeroute_2.3.22.tgz
A copy should also be available in a folder on your Freedom Fone server desktop

To upload a configuration backup to your OfficeRoute, go to *Administration → Configuration backup*. Click on the *Configuration backup* link and select *Upload config* from the submenu.

Browse for the OfficeRoute configuration file on your computer; leave “Don't extract licence” checkbox ticked. Click the Upload button in the bottom right hand corner of the screen.



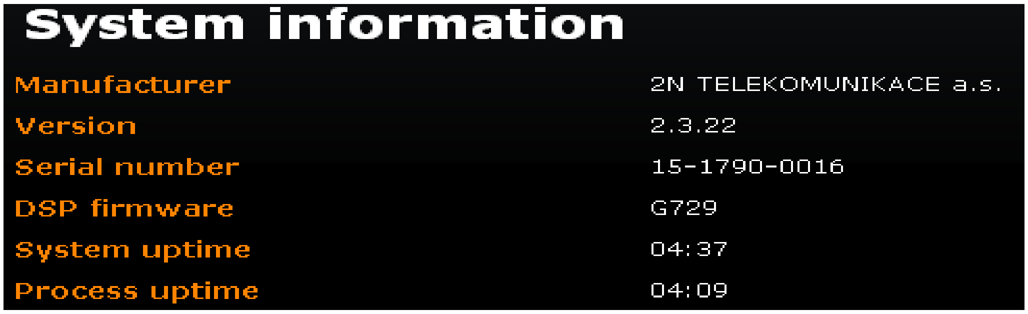
Upload config

Upload configuration from file — No file selected.

Don't extract licence ☑

Note

For the *Upload config* to work, the OfficeRoute firmware should be v2.3.22. This information can be seen at the bottom of the *Network* page. If you need to update the firmware on your OfficeRoute, download the latest version from http://www.2n.cz/en/downloads/p=officeroute|category=firmware

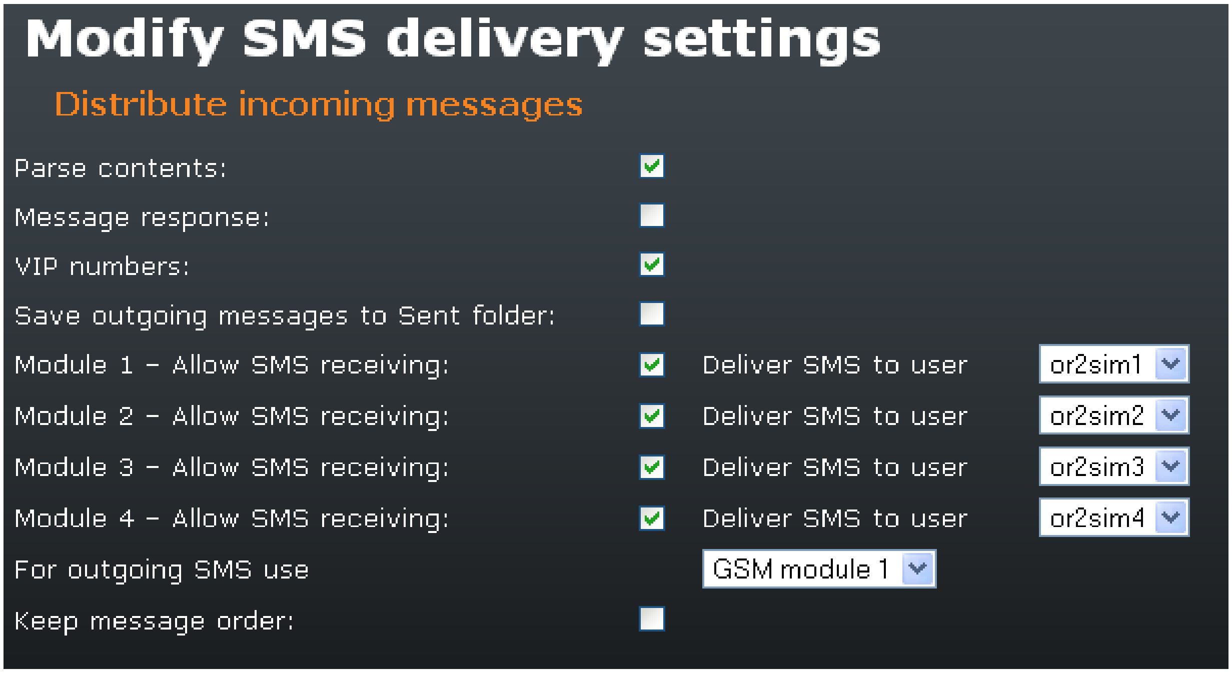


System information

Manufacturer	2N TELEKOMUNIKACE a.s.
Version	2.3.22
Serial number	15-1790-0016
DSP firmware	G729
System uptime	04:37
Process uptime	04:09

Go to *Administration* → *Update firmware* to update the firmware on your OfficeRoute.

SMS delivery settings for the distribution of SMSs into different Inboxes for each sim card will look as follows:



Modify SMS delivery settings

Distribute incoming messages

Parse contents: ☑

Message response: □

VIP numbers: ☑

Save outgoing messages to Sent folder: □

Module 1 – Allow SMS receiving: ☑ Deliver SMS to user or2sim1

Module 2 – Allow SMS receiving: ☑ Deliver SMS to user or2sim2

Module 3 – Allow SMS receiving: ☑ Deliver SMS to user or2sim3

Module 4 – Allow SMS receiving: ☑ Deliver SMS to user or2sim4

For outgoing SMS use GSM module 1

Keep message order: □

10 Freedom Fone and SIP

The default policy of Freedom Fone is to only allow authenticated SIP calls. However, in order to allow unauthorised users to call into Freedom Fone via SIP (which is free of charge and only requires an Internet connection), we have implemented a solution which enables this.

Furthermore, to block VoIP scanners and other unsolicited traffic to hit the platform, we have implemented a firewall that blocks unwanted traffic.

10.1 Non authenticated SIP calls

SIP authentication can be set (true|false) in the configuration file core.php

```
/opt/freedomfone/gui/app/Config/core.php

define ('NO_SIP_AUTH',false);
```

true: No SIP authentication will take place. This will allow you do make unauthenticated SIP calls to Freedom Fone to reach its voice services.

false: SIP authentication is required.

The NO_SIP_AUTH setting is reflected in the dialplan.

```
/opt/freedomfone/xml_curl/dialplan.xml
```

10.2 Firewall to block unwanted traffic

If you set the NO_SIP_AUTH to ***true***, you allow any IP address to make a non-authenticated call to your platform. This is frequently misused by so called VoIP scanners, robots that aim to establish free phone calls though poorly configured VoIP gateways. VoIP scanners perform password brute force attacks, which generate large numbers of CS_DESTROY records. The CS_DESTROY records are shown in the Freedom Fone GUI as incomplete calls, and gives a misleading image of incoming calls.

To deal with the problem of unwanted traffic, a firewall has been implemented that blocks all SIP registration traffic that comes from any network that is not included in the whitelist.

The firewall is managed by a bash script (/opt/freedomfone/firewall/run.sh) which allows the following options:

```
/opt/freedomfone/firewall/run.sh {config|show|start|stop}

      config: setup the whitelist file (/opt/freedomfone/firewall/whitelist.txt)
      show:   shows status of firewall (whitelist and iptables rules)
      start:  add rules from the iptables firewall
      stop:   remove rules from the iptables firewall
```

Example setup

This example will set up a firewall that only allows SIP calls from the networks 192.168.1.0/24 and 192.168.2.0/24.

Step 1) Configuration wizard

When running the firewall script with the config option, the user will be guided through a configuration wizard which will create a whitelist. Remember to run commands as `sudo su`

```
root@sharicus:/opt/freedomfone/firewall# bash run.sh config

What is the public interface of your freedomfone installation? eth0
Do you want  to be able to place calls in your freedomfone installation (y/n)? y
Do you want 192.168.1.0/24 to able to place calls in your freedomfone installation
(y/n)? y
Do you want to allow more networks to place calls in your freedomfone installation
(y/n)? y
Enter network in CIDR format e.g. 192.168.0.0/16: 192.168.2.0/24
Do you want to allow more networks to place calls in your freedomfone installation
(y/n)? n
The following networks will be whitelisted in the firewall

192.168.1.0/24
192.168.2.0/24
```

For help with CIDR formats see http://www.ipaddressguide.com/

Step 2) Add rules to firewall

This step creates the iptables rules, and must be run after creating the whitelist.

```
root@sharicus:/opt/freedomfone/firewall# bash run.sh start
```

Step 3) Show firewall status

The final step shows what settings that have been made. Please note that this option takes some time to display the result.

```
root@sharicus:/opt/freedomfone/firewall# bash run.sh show

Networks whitelisted

192.168.1.0/24
192.168.2.0/24

Chain INPUT (policy ACCEPT)
target     prot opt source               destination
ACCEPT     udp  --  192.168.0.0/16       anywhere             udp dpt:sip
ACCEPT     udp  --  192.168.0.0/16       anywhere             udp dpt:5080
LOG        udp  -- !192.168.0.0/16       anywhere             udp dpt:sip LOG level
warning prefix "Freedom Fone SIP registration"
LOG        udp  -- !192.168.0.0/16       anywhere             udp dpt:5080 LOG
level warning prefix "Freedom Fone SIP registration"
ACCEPT     udp  --  192.168.1.0/24       anywhere             udp dpt:sip
ACCEPT     udp  --  192.168.1.0/24       anywhere             udp dpt:5080
LOG        udp  -- !192.168.1.0/24       anywhere             udp dpt:sip LOG level
warning prefix "Freedom Fone SIP registration"
LOG        udp  -- !192.168.1.0/24       anywhere             udp dpt:5080 LOG
level warning prefix "Freedom Fone SIP registration"
DROP       udp  --  anywhere             anywhere             udp dpt:sip
DROP       udp  --  anywhere             anywhere             udp dpt:5080
ACCEPT     udp  --  192.168.1.0/24       anywhere             udp dpt:sip
ACCEPT     udp  --  192.168.1.0/24       anywhere             udp dpt:5080
LOG        udp  -- !192.168.1.0/24       anywhere             udp dpt:sip LOG level
warning prefix "Freedom Fone SIP registration"
LOG        udp  -- !192.168.1.0/24       anywhere             udp dpt:5080 LOG
level warning prefix "Freedom Fone SIP registration"
ACCEPT     udp  --  192.168.2.0/24       anywhere             udp dpt:sip
ACCEPT     udp  --  192.168.2.0/24       anywhere             udp dpt:5080
LOG        udp  -- !192.168.2.0/24       anywhere             udp dpt:sip LOG level
warning prefix "Freedom Fone SIP registration"
LOG        udp  -- !192.168.2.0/24       anywhere             udp dpt:5080 LOG
level warning prefix "Freedom Fone SIP registration"
DROP       udp  --  anywhere             anywhere             udp dpt:sip
DROP       udp  --  anywhere             anywhere             udp dpt:5080

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
```

10.3 Logs

All SIP registrations are logged in:

```
 /var/log/syslog
```

11 Log files

Freedom Fone stores its logs in two different locations depending on which component is concerned.

Case 1: Freedom Fone generated logs
Location: /opt/freedomfone/gui/app/tmp/logs

Case 2: Logs generated by other components (for example web server, dispatcher, Gammu, backend sweeper, pop3 daemon (incoming sms from OfficeRoute)).
Location: /opt/freedomfone/log

The postfix log is located in /var/log/mail.log

All Freedom Fone logs can be read by the Admin user from the Freedom Fone GUI (*Dashboard* → *Logs*).