Icecom SwitchBoard

Version 2.0

Asterisk Configuration Manual

Icecom SwitchBoard software is based on technology developed by Icecom Ltd. Icecom reserves the right to make changes in the content of the software. The company is not obliged to give notification of any possible changes. Icecom accepts no liability for any possible errors in the software. Icecom is not responsible for the operability of the software with all kinds of hardware; nor does it guarantee protection against possible computer viruses or any attempts against security.

This manual was made for the Asterisk configuration for Icecom SwitchBoard version 2.0. Icecom Ltd. reserves the right to change the content of the publication without special notice.
The names and phone numbers used in the publication are fictional.

PostgreSQL Database Management System
(formerly known as Postgres then as Postgres95)

CONTENTS

INTRODUCTION

This configuration manual was made for Icecom SwitchBoard version 2.0, for configuring Asterisk. The starting point for the manual is that the Icecom SwitchBoard software and Asterisk be already installed on the computer. The installation instructions for the Icecom SwitchBoard software can be found in a separate manual.

This document explains what the Asterisk server configuration files must include in order for the Asterisk message to be transmitted to the Icecom SwitchBoard.

All the files to be configured can be found in the file ***/etc/asterisk***.

In the manual, text marked with the Verdana font signifies lines to be entered in the configuration files. Computer-specific variables for the configuration have been marked inside tags <> and comment lines have been marked to start with a semicolon ;. In addition, the names of files and directories have been written *in italics*.

At the end of the document, there are the example files for a configuration, in which there are 20 extensions besides the switchboard attendant (appendices 1 – 6)

Terms used commonly in the manual:

context	*extensions.conf* file element marked inside square brackets. A context ends when new square brackets begin.
extension (exten)	A function marked for a single extensions in the *extensions.conf* file. An extension begins with the character sequence ”exten =>”, which is followed by an ID identifying the extension and then the priority (=executing order) and function divided by a comma. (for example exten => 200,1,Dial(SIP/200))

1 MANAGER.CONF

The Asterisk is given the direction to use the manager interface from port 5038.

```
[general]
enabled = yes
port = 5038
```

Add user manager, whose username is **login** and password is **passwd** (**Note!** Username and password must not be changed!), and the rights to use the functions set:

```
[login]
secret = passwd
read = system,call,log,verbose,command,agent,user
write = system,call,log,verbose,command,agent,user
```

2 EXTENSIONS.CONF

2.1 Setting up a common prefix to supplement extensions

A common prefix for the extensions is set in the beginning of the file in [**globals**] context. The prefix is set as the variable ”start”.

```
[globals]
start=<common prefix for extensions>
; the common prefix can be for example 5277
```

The prefix is used later when setting an extension for placing an outside call (chapter 2.4.6).

2.2 Setting up queue music

The accompanying three files can be set as the queuing music:

- *fur-elise.gsm*
- *turkishm.gsm*
- *bourree-pizzicato.gsm*

The files must be saved in the directory */var/lib/asterisk/sounds/***.**

For the queuing music, extensions are created in the file’s [**default**] context.
There are four extensions:

- **music**: for incoming calls queue
- **hold**: for queued calls queue
- **nite**: for night switching
- **busy**: for busy line queue

Any personal messages/recordings saved in the system must be files of the .gsm type, and they must reside in the directory */var/lib/asterisk/sounds*.

```
[default]
exten => music,1,Answer
exten => music,2,Playback(you_are_in_the_queue)
;       you_are_in_the_queue-wait_a_moment: own recording
exten => music,3,Playback(fur-elise)
exten => music,4,Playback(turkishm)
exten => music,5,Playback(bourree-pizzicato)
exten => music,6,Goto(music|3)

exten => hold,1,Answer
exten => hold,2,Playback(receiver_busy)
;       receiver_busy: own recording
exten => hold,3,Goto(music|3)
;       caller is put on hold

exten => nite,1,Answer
exten => nite,2,Playback(we_are_open_from_8_to_16)
;       we_are_open_from_8_to_16: own recording
exten => nite,3,Hangup

exten => busy,1,Answer
exten => busy,2,Playback,(person_is_busy)
;       person_is_busy: own recording
exten => busy,3,Hangup
exten => busy,400,Busy
```

2.3 Setting up attended transfers between external calls

When wanting to support attended transfer between two external calls in the system, an extension named **conference** must be created in the file's [**default**] context.

```
exten => conference,401,Meetme(1|q)
exten => conference,501,Meetme(2|q)
exten => conference,601,Meetme(3|q)
exten => conference,701,Meetme(4|q)
```

Please note!
The priorities of the ”conference” extension must be the same as the priorities set in the meetme table in the swb database of postgresql. By default, four priorities have been set up in the database: 401, 501, 601 and 701.

2.4 Setting up extensions

Extensions can be set up for two different types of users:

- switchboard attendants
- power users

All extensions (except macro) are placed under the [**default**] context by default.

2.4.1 Extension for switchboard attendant

There are two lines in a switchboard attendant's extension. On the first one the attendant's phone rings. On the second one, when the attendant is busy (priority 102), the call is transferred to a call queue titled "music". (Defined in chapter 2.2).

Example 1. Extension for two switchboard attendants (extensions 201 and 202):

```
exten => 201,1,Dial(SIP/${EXTEN})
exten => 201,102,Goto(music|2)
;
exten => 202,1,Dial(SIP/${EXTEN})
exten => 202,102,Goto(music|2)
```

2.4.2 Extension for virtual switchboard attendant

When calling a joint number for two switchboard attendants, both their extensions will ring at the same time. The numbers must be defined as explained in chapter 2.4.1. Then an individual physical phone device is not required for the joint number.

Example 2. When calling joint number 200, extensions 201 and 202 ring:

```
exten => 200,1,Dial(SIP/201&SIP/202)
exten => 200,102,Goto(music|2)
```

2.4.3 Extension for power user

A power user's extension is defined by one user-specific line, stating the user's extension number, referring to the common macro between the users and defining other user-specific values.

Example 3. Defining a power user's extension:

```
exten => 203,1,Macro(sipcall,20,200)
```

In example 3, the user's extension number is 203 and sipcall refers to the macro defined in example 4. 20 is the time that the phone rings (in example 4 $ {ARGS1}), and 200 the number to which the call is transferred if the number is busy in example 4 ${ARGS2}).

2.4.4 Macro context shared by all power users

A macro is placed into its own context, where its name is put in square brackets [ ] and started by the prefix **macro**.

Example 4. Macro named sipcall:

```
[macro-sipcall]
exten => s,1,Dial(SIP/${MACRO_EXTEN}|${ARGS1})
exten => s,2,Wait(1)
exten => s,3,Voicemail(u${MACRO_EXTEN} )
;       alternatively queue transfer can be used:
;       exten => s,3,Goto(default, ${ARGS2},1)
exten => s,102,Wait(1)
exten => s,103,Goto(default, ${ARGS2},1)
;       default means the [default] context defined in the file, where
;       the switchboard attendant's extension has been set by default
```

2.4.5 Attended transfer to internal number

The extension for an attended transfer is presented with a single line. The first number to be placed after the underline character refers to the first number of the whole extension group. In example 5, 2XX refers to extensions 200 – 299 (i.e. the number sequence for extensions 100 – 199 would be 1XX, for extensions 300 – 399 3XX, etc.)

Example 5. Attended transfer set for extensions 200 – 299:

```
exten => _2XX,401,Meetme(${EXTEN}|q)
```

2.4.6 Extension for a call placed outside

The extension defined for a call to be placed outside is marked with the **_XXX.** symbol sequence **(Note!** a dot at the end of the extension). Then the number to be called must be at least 3 digits long and may contain the numbers 0 – 9.

In example 6, it is first checked if the call is placed from Icecom SwitchBoard, and then the caller name is changed if necessary. The actual call is placed in priorities 4 or 8. In priority 400, the call is transferred to the switchboard attendant, if it could not be made in the previous stage. In the example, the variable ”start” is a four-character sequence (see chapter 2.1), and ”CALLERIDNUM:0:4” is marked in priority 6. If the length of the variable is something else, the length is marked in priority 6.

Example 6. Call placed outside:

```
exten => _XXX.,1,GotoIf($[${CALLERIDNUM} = asteriskCaller]?2:6)
exten => _XXX.,2,AGI,Lookup.agi  ;see chapter 2.4.7
exten => _XXX.,3,SetCallerId,${start}${CIDNUM}  ;start: see chapter 2.1
exten => _XXX.,4,Dial,Zap/g1/${EXTEN}
exten => _XXX.,5,Goto(${EXTEN}|400)
exten => _XXX.,6,GotoIf($[${CALLERIDNUM:0:4} = ${start}]?8:7)
exten => _XXX.,7,SetCallerId,${start}${CALLERIDNUM}
exten => _XXX.,8,Dial,Zap/g1/${EXTEN}
exten => _XXX.,9,Goto(${EXTEN}|400)
exten => _XXX.,400,Goto(200|1)
```

2.4.7 Lookup.agi for outgoing calls

The Lookup.agi script changes the caller's callerId from one called asteriskCaller into the caller's own, when the call is placed from the Icecom SwitchBoard. If this script is not executed, asteriskCaller is shown as the callerId of all calls started from the Icecom SwitchBoard.

Lookup.agi and Lookup.class and files must be copied to directory /var/lib/asterisk/agi-bin/. Switchboard.properties file must be copied to directory /etc/asterisk/. Switchboard.properties file includes Icecom SwitchBoard's internal user agent's name (asteriskCaller by default). Please check the access rights for the files (required commands are "chmod a+rxw Lookup.agi" and "chmod a+rxw Lookup.class"). These three files can be downloaded from the download-section of Icecom SwitchBoard website (www.icecomswitchboard.com).

Check JAVA_HOME variable from Lookup.agi –file.

3 SIP.CONF

Create an own context for Icecom SwitchBoard's internal user agent (by the same name as set in the database, asteriskCaller by default).

```
[asteriskCaller]
type=friend
host=dynamic
insecure=yes
```

An individual context must be created for every user (switchboard attendants and power users), with the user's extension number as the context name. The variable "secret" is a password with which the user's phone registers to Asterisk.

```
[name]
type = friend
secret = password
host = dynamic
canreinvite= yes
```

Please note!
The value canreinvite=yes must be set for all users.

When wanting to change the name of the user agent called asteriskCaller into another one, the same change must be made also to the asterisksettings table of the postgresql swb database.

4 CDR_PGSQL.CONF

In the file *Cdr_pgsql.conf*, insert the IP address where the database resides and the super-user set up for it. By default the username is postrgres, with **no password**. The name of the database is *swb*.

```
[global]
hostname=<database IP address>
dbname=swb
password=
user=postgres
port=5432
```

This configuration gives the Asterisk the command to use cdr table in the PostgreSQL database called *swb*.

5 MEETME.CONF

For attended transfer, add a conference number for every user to the *meetme.conf* file.
(See user priority in example 5.)

Example 7. Add user extension numbers 203 and 204:

```
conf => 203
conf => 204
```

Example 8. Add own attended transfer numbers for calls coming in from the outside:

```
conf => 1
conf => 2
conf => 3
conf => 4
```


6 MODULES.CONF

The use of the PostgreSQL database is added to the modules used by the Asterisk server. Existing configurations must be kept in the file and the following ones added:

```
[modules]
load => cdr_pgsql.so
...

[global]
cdr_pgsql.so=yes
...
```


APPENDIX 1. Example of Asterisk extensions.conf file configuration (2 switchboard attendants and 18 (basic) extensions)

```
[globals]
start=5277
;
[general]
static=yes
writeprotect=no
;
[local]
include => default
;
[default]
;
;           QUEUE WHEN CALLING FROM THE OUTSIDE
exten => music,1,Answer
exten => music,2,Playback(please_wait)                    ; own recording
exten => music,3,Playback(fur-elise)
exten => music,4,Playback(turkishm)
exten => music,5,Playback(bourree-pizzicato)
exten => music,6,Goto(music|3)
;
;           INTERNAL QUEUE
exten => hold,1,Answer
exten => hold,2,Playback(vm-theperson)
exten => hold,3,Playback(vm-isonphone)
exten => hold,4,Goto(music|3)
;
;           NIGHT SWITCHING
exten => nite,1,Answer
exten => nite,2,Playback(we_are_open_from_8_to_16)        ; own recording
exten => nite,3,Hangup
;
;           PERSON BUSY
exten => busy,1,Answer
exten => busy,2,Playback(person_is_busy)                  ; own recording
exten => busy,3,Hangup
exten => busy,400,Busy
;
;           TIMEOUT AND INVALID OPTIONS
exten => t,1,Playback(invalid)
exten => i,1,Goto(music|3)
;           CALLING OUTSIDE, CALLERID IS CHECKED
;           (start=common extension prefix and 200 = number for shared switchboard)
exten => _XXX.,1,GotoIf($[${CALLERIDNUM} = asteriskCaller]?2:6)
exten => _XXX.,2,AGI,Lookup.agi
exten => _XXX.,3,SetCallerId,${start}${CIDNUM}
exten => _XXX.,4,Dial,Zap/g1/${EXTEN}
exten => _XXX.,5,Goto(${EXTEN}|400)
exten => _XXX.,6,GotoIf($[${CALLERIDNUM:0:4} = ${start}]?8:7)
exten => _XXX.,7,SetCallerId,${start}${CALLERIDNUM}
exten => _XXX.,8,Dial,Zap/g1/${EXTEN}
exten => _XXX.,9,Goto(${EXTEN}|400)
exten => _XXX.,400,Goto(200|1)
;
```


```
;           ATTENDED TRANSFER FOR EXTERNAL NUMBERS
exten => conference,401,Meetme(111|q)
exten => conference,501,Meetme(222|q)
exten => conference,601,Meetme(333|q)
exten => conference,701,Meetme(444|q)
;
;           SHARED SWITCHBOARD – NO PHYSICAL PHONE DEVICE
exten => 200,1,Dial(SIP/201&SIP/202)
exten => 200,2,Goto(music|1)
exten => 200,102,Goto(music|2)
;
;           SWITCHBOARD 1         - extension 201
exten => 201,1,Dial(SIP/${EXTEN})
exten => 201,102,Goto(music|2)
exten => 201,401,Meetme,201|q
;
;           SWITCHBOARD 2         - extension 202
exten => 202,1,Dial(SIP/${EXTEN})
exten => 202,102,Goto(music|2)
exten => 202,401,Meetme,202|q
;
;           POWER USERS           - extensions 203 – 220
exten => 203,1,Macro(sipcall,20,200)
exten => 204,1,Macro(sipcall,20,200)
exten => 205,1,Macro(sipcall,20,200)
exten => 206,1,Macro(sipcall,20,200)
exten => 207,1,Macro(sipcall,40,200)
exten => 208,1,Macro(sipcall,20,200)
exten => 209,1,Macro(sipcall,20,200)
exten => 210,1,Macro(sipcall,20,200)
exten => 211,1,Macro(sipcall,20,200)
exten => 212,1,Macro(sipcall,40,200)
exten => 213,1,Macro(sipcall,20,200)
exten => 214,1,Macro(sipcall,20,200)
exten => 215,1,Macro(sipcall,20,200)
exten => 216,1,Macro(sipcall,60,200)
exten => 217,1,Macro(sipcall,20,200)
exten => 218,1,Macro(sipcall,10,200)
exten => 219,1,Macro(sipcall,20,200)
exten => 220,1,Macro(sipcall,20,200)
;
;           ATTENDED TRANSFER FOR INTERNAL NUMBERS
exten => _2XX,401,Meetme(${EXTEN}|q)
;
;           JOINT MACRO CONTEXT FOR POWER USERS
[macro-sipcall]
exten => s,1,Dial(SIP/${MACRO_EXTEN}|${ARGS1})
exten => s,2,Voicemail(u${MACRO_EXTEN})
; exten => s,2,Goto(default,${ARGS2},1)
exten => s,102,Goto(default,${ARGS2},1)
;
```


APPENDIX 2. Example of sip.conf file configuration

```
;
; SIP Configuration for Asterisk
;
[general]
port = 5060                     ; Port to bind to
context = default               ; Default for incoming calls
maxexpirey=3600                 ; Max length of incoming registration allowed
allow=ulaw                      ; Allow codecs in order of preference
;
[asteriskCaller]
type=friend
insecure=yes
host=dynamic
;
[201]
type=friend
secret = password1
host=dynamic
canreinvite=yes
;
[202]
type=friend
secret = password2
host=dynamic
canreinvite=yes
;
[203]
type=friend
secret = password3
host=dynamic
canreinvite=yes
;
[204]
type=friend
secret = password4
host=dynamic
canreinvite=yes
;
[205]
type=friend
secret = password5
host=dynamic
canreinvite=yes
;
[205]
type=friend
secret = password6
host=dynamic
canreinvite=yes
;
[207]
type=friend
secret = password7
host=dynamic
canreinvite=yes
;
```


```
[208]
type=friend
secret = password8
host=dynamic
canreinvite=yes
;
[209]
type=friend
secret = password 9
host=dynamic
canreinvite=yes
;
[210]
type=friend
secret = password 10
host=dynamic
canreinvite=yes
;
[211]
type=friend
secret = password 11
host=dynamic
canreinvite=yes
;
[212]
type=friend
secret = password 12
host=dynamic
canreinvite=yes
;
[213]
type=friend
secret = password 13
host=dynamic
canreinvite=yes
;
[214]
type=friend
secret = password 14
host=dynamic
canreinvite=yes
;
[215]
type=friend
secret = password 15
host=dynamic
canreinvite=yes
;
[216]
type=friend
secret = password 16
host=dynamic
canreinvite=yes
;
[217]
type=friend
secret = password 17
host=dynamic
canreinvite=yes
;
```


```
[218]
type=friend
secret = password 18
host=dynamic
canreinvite=yes
;
[219]
type=friend
secret = password 19
host=dynamic
canreinvite=yes
;
[220]
type=friend
secret = password 20
host=dynamic
canreinvite=yes
;
```

APPENDIX 3. Example of meetme.conf file configuration

```
;
[rooms]
conf => 200
conf => 201
conf => 202
conf => 203
conf => 204
conf => 205
conf => 206
conf => 207
conf => 208
conf => 209
conf => 210
conf => 211
conf => 212
conf => 213
conf => 214
conf => 215
conf => 216
conf => 217
conf => 218
conf => 219
conf => 220
;
conf => 1
conf => 2
conf => 3
conf => 4
;
```


APPENDIX 4. Example of cdr_pgsql.conf file configuration

```
;
[global]
hostname=213.255.100.10                    ; change IP address
dbname=swb
password=
user=postgres
port=5432
;
```

APPENDIX 5. Example of modules.conf file configuration

```
;
; Asterisk configuration file
;
; Module Loader configuration file
;
[modules]
autoload=yes
noload => pbx_gtkconsole.so
noload => pbx_kdeconsole.so
noload => app_intercom.so
noload => chan_oss.so
load => cdr_pgsql.so
load => res_parking.so
;
[global]
cdr_pgsql.so=yes
;
```

APPENDIX 6. Example of manager.conf file configuration

```
;
; Asterisk Call Management support
;
[general]
enabled = yes
port = 5038
;
[login]
secret=passwd
read=system,call,log,verbose,command,agent
write=system,call,log,verbose,command,agent
;
```