

## [telemarketer]

This is an interesting application that can stop most telemarketers dead in their tracks.

I'll go over the most pertinent lines to explain what it does.

[telemarketer] is the name of the context. Typically you would point your incoming phone line or trunks to this context.

The first line **exten => s,1,Answer()** answers the line then the second line waits for 2 seconds to allow the connection to settle. Often cell phones tend to take a second or two for the audio to cut through.

Then it drops through to the

```
exten=>s,n,GoToIf($[${CALLERID(num)} = 5203211234]?allow)
```

The (CALLERID)(num) function picks up the phone number calling in and if it matches, the ?allow sends it to the

```
exten => s,n(allow),Goto(cohncom,6103,1)
```

which tells it to go to the context called cohncom and ring extension 6103.

**Note:** Use the appropriate phone numbers to allow them to ring directly through.

What this is all about is you can provide a list of “friends and family” that are allowed to call you and their calls will ring directly to the specified phone.

Now, take a look at what happens if the phone number doesn't match.

It falls through to the **exten => s,n,Zapateller()** line which plays a SIT (Special Information Tone) which usually precedes a recording saying that number is not in service or has been disconnected..

Very often, the telemarketers equipment will mark that number as not in service and not call it again.

The next line, **exten => s,n,Background(telemarketer)** tells Asterisk to play a message called telemarketer. This message is not a standard sound file and you will have to record it yourself or download the one I had Allison Smith record.

While playing the message, the caller has a few seconds to press the number specified in the message to ring through. This allows legitimate callers who are not on your “friends and family” list to contact you after hearing the message.

In this case, the number is 4 so it goes to the context cohncom and rings extension 6103 as before.

Many people have learned that if you hit 0 (zero) when a message plays, it will often connect you to a live person. We want to make sure that only the correct digit will work.

The **exten => i,1,Playback(pbx-invalid)** will play a message saying that number is incorrect, please try again and returns them to **exten => s,1,Answer()** in the context telemarketer making them start all over again unless they press only the digit(s) you specify.

Finally we need to deal with people who just stay on the line as in “If you have a rotary phone, please stay on the line.”

Hopefully, by now most of your land line callers will have sprung a few dollars and moved into the 21st century with a DTMF (touchtone ®) phone.

Otherwise, the **exten => t,1,Playback(vm-goodbye)** will just time them out and hang up after a few more seconds.

However, if your cheap land line friends let you know, you can add their phone number to your list and even a rotary phone will ring through.

The reason I came up with this little script is that even though my home phone and cell phones are on the NATIONAL DO NOT CALL REGISTRY <http://www.donotcall.gov> list, many of the telemarketers now seem to be coming in from countries like India and apparently they are not required to obey the Do Not Call restrictions.

Any telemarketer who chooses to ring through after hearing the message usually gets a very chilly reception followed by comments about their IQ and/or their parental ancestry!

[telemarketer]

exten => s,1,Answer()

exten => s,n,Wait(2)

exten => s,n,GoToIf(\${CALLERID(num)} = 5203211234]?allow) ; Good list.

exten => s,n,Zapateller() ; Plays SIT tone

exten => s,n,Background(telemarketer) ; Plays nasty message about who is allowed

exten => s,n,WaitExten(5) ; to call.

exten => 4,1,Goto(cohncom,6103,1) ; Same by pressing 4

exten => i,1,Playback(pbx-invalid) ; Caller pressed wrong digit

exten => i,n,Goto(telemarketer,s,1) ; Gets to start all over again

exten => t,1,Playback(vm-goodbye) ; Waited too long or rotary phone

exten => t,n,Hangup() ; Spend 10 bucks and join 21st Century

exten => s,n(allow),Goto(cohncom,6103,1) ;If on good list rings this phone immediately

---

This is the scenario to allow using a database to hold the list of allowed callers. You have to install asterisk-mysql unixODBC unixODBC-dev for asterisk to be able to communicate with a mysql database. There is a good chapter on this in the 3<sup>rd</sup> edition of the Asterisk Future of Telephony book as well as on-line at

[http://www.asteriskdocs.org/en/3rd\\_Edition/asterisk-book-html-chunk/installing\\_configuring\\_odbc.html](http://www.asteriskdocs.org/en/3rd_Edition/asterisk-book-html-chunk/installing_configuring_odbc.html)

Once that is configured, we create a mysql database called telemarketer which is owned by asterisk with a password of Cis218.

There are 5 fields in the table called phonenumber.

1 is the first name

2 is the last name

3 is type (work, cell, home, etc.)

4 is number which is a 10 digit phone number. IE: 6025551212 (Note: No spaces or dashes)`

5 is just a flag called allow and the default value is 1

If your phone number is in the database, this script works and sends you to the proper context of cohncom extension 6103.

If your phone number is not in the database, you fall through to the telemarketer context which plays the zapateller tone and a nasty recorded message.

The problem is it also generates a mysql error in asterisk because it doesn't find a number in allow about 9 lines down in this context to satisfy the mysql syntax.

The question is for someone who knows more about mysql than I as to a method to return a value of 0 for allow if the number is not in the database at all which would be the normal situation.

Maybe using COALESCE or IFNULL?

```
[mysql_test2]
```

```
exten => s,1,Answer
```

```
exten => s,n,Set(CallerIDString=${CALLERID(num)})
```

```
exten => s,n,MYSQL(Connect connid localhost asterisk Cis218 telemarketer)
```

```
exten => s,n,mysql(Query resultid ${connid} SELECT number, allow FROM phonenumbers WHERE number=${CallerIDString})
```

```
exten => s,n,mysql(Fetch fetchid ${resultid} number allow)
```

```
exten => s,n,NoOp("We found: ${number} ${allow}"); Just a test to see what we got from incoming caller
```

```
exten => s,n,mysql(Clear ${resultid})
```

```
exten => s,n,mysql(Disconnect ${connid});
```

```
exten => s,n,GoToIf($[${allow} = 1]?cohncom,6103,1)
```

```
exten => s,n,Goto(telemarketer,s,1)
```

```
exten => s,n,Hangup()
```