

# Asterisk Caller-ID How-to

January 4, 2006

## Abstract

This How-To shows how a simple shell script can be used with Asterisk to set the name of an incoming caller based on the transmitted phone number. Although this is no *enterprise*-solution, it is a portable solution when limited resources (memory / speed) are an issue.

## 1 Introduction

This is the first article in the series of mini How-to articles on configuring Asterisk on limited resource devices such as OpenWRT-routers. Running on a device with limited resources (cpu-speed / memory), part of the problem is to use only the standard tools available, especially that are part of the commands provided by the *busybox*. Focusing on these basic commands also ensures that the solution will most likely run on other systems as well.

This article shows how a simple AGI script can use a flat-file database to set the Caller-ID. This solution has been tested on a Linksys WRT54GS v. 1.1 running OpenWRT *white russian rc4* and the *asterisk-cvs* package. It cannot be considered to be *enterprise-grade*. If you are looking for one, look at a LDAP based solution instead. When you need help implementing such a solution, feel free to contact the author.

## 2 Dial-plan

All incoming calls are handled in the `extensions.conf` (usually located in `/etc/asterisk/`). To set the caller-ID name, we need to call our shell-script with the phone number to receive the caller name and then set the caller information using the Asterisk command `SetCallerID`. The commands in the `extensions.conf` should therefore be

```
exten => s,n,AGI,cidlookup.agi|${CALLERIDNUM}
exten => s,n,SetCallerID(${LONGNAME}<${CALLERIDNUM}>)
```

The value of the variable `${CALLERIDNUM}` is set automatically by Asterisk and is passed on to our `cidlookup.agi` script. The `astagidir`-variable in the `asterisk.conf` tells Asterisk in which directory to look for the script. The default value on the tested OpenWRT system is `/usr/lib/asterisk/agi-bin`.

## 3 Shell-script

We use a flat file named `cid.txt` to store the names and phone numbers. Each line in the file contains the numbers of one person. A single line looks like this:

```
NAME##TELEINFO1:NUMBER1##TELEINFO2:NUMBER2##...
```

where `NAME` is the name as it should appear in the caller-ID, `TELEINFO` is an addition as *Home*, *Work* or *Mobile* we use to differ the various phone numbers of a person. This *database* file is used by the shell-script file `cidlookup.agi` itself.

```
#!/bin/ash
grep "\:$1##" /usr/lib/asterisk/agi-bin/cid.txt | head -n 1 | \
awk -F"##" 'BEGIN {OFS=""}{print "SET VARIABLE LONGNAME \"\",$1,\"\"}'
exit 0
```

Because this shell script is sensitive to trailing *whitespace*, ensure that the line ends right after the *backspace* (`\`). The `head` command ensures that only the first match is returned.<sup>1</sup> In the `awk` command the `-F"##"` option changes the field separators and the `OFS` setting eliminates any spaces around the matched values.

## 4 Database-File

There are numerous ways on how to create the database file needed. When using the Address-book of an Macintosh, an easy way to start, would be to using the following AppleScript:<sup>2</sup>

```
tell application "Address Book"
  set liste to ""
  repeat with p in people
    set pline to ""
    repeat with t in phone of p
      set fnumber to ""
      set pline to pline & label of t & ":" & value of t as text & "##"
    end repeat
    if pline is not "" then
      set fname to first name of p & " " & last name of p
      if company of p then
        set fname to organization of p
      end if
      set liste to liste & fname & "##" & pline & return
    end if
  end repeat
  liste
end tell
```

## Disclaimer

Please contact the author in case you find any errors or think of any ways on how to improve this paper.

Although the information and code has been edited carefully and seems to work as expected on our machine with our data, please be advised that any use of it is at your own risk and without any warranty. Using the information / code given in this paper may result in loss of data as well as damage and (costly) malfunction of your equipment.

<sup>1</sup>The `grep` command on the *busybox* is missing the `'-m'` option to limit the number of matches.

<sup>2</sup>More advanced versions of this script can be obtained from the author.