

Asterisk

Content migrated from [here](#).

Asterisk Download Links

<http://downloads.asterisk.org/pub/telephony/>

<http://downloads.asterisk.org/pub/telephony/dahdi-linux/>

<http://downloads.asterisk.org/pub/telephony/dahdi-tools/>

<http://downloads.asterisk.org/pub/telephony/libpri/>

<http://downloads.asterisk.org/pub/telephony/asterisk/>

Asterisk SVN Links

<http://svnview.digium.com/svn>

<http://svnview.digium.com/svn/asterisk/>

Asterisk Related Links

- [Asterisk snmp](#)
- [Asterisk 1.4 app_page.c added device state](#)
- [Asterisk & rtcp](#)
- [Asterisk :: The Open Source Telephony Platform](#)
- [Building asterisk](#)

Telephony Related Links

- [Analog Voice Fundamentals](#)
- [Polycom](#)
- [T1 Channel Numbering](#)

Hacks and Patches

- [Logrotate configuration](#)
- [Music on hold](#)
- [Voicemail VM_ALTEXT Patch](#)

Converting audio files for use in Asterisk

Resample a stereo wav into a mono 8000Hz 16 bit wav:

```
sox infile.wav -c mono -r 8000 -b 16 outfile.wav
```

Convert a properly formatted wav (8000Hz, 16bit mono) file to gsm:

```
sox infile.wav outfile.gsm
```

Converting mp3 to slin

```
mpg123 -w file.wav file.mp3
```

```
sox file.wav -t raw -r 8000 -s -2 -c 1 file.sln
```

Converting wav to slin

```
sox file.wav -t raw -r 8000 -s -2 -c 1 file.sln
```

Upgrading asterisk 1.4 to 1.6

NoOp updates

/NoOp

```
:%s/\(NoOp,\)\(.*\)$/NoOp(\2)/
```

```
sed -e 's/\(NoOp,\)\(.*\)$/NoOp(\2)/'
```

pipe character to comma

```
:%s|/,/g
```

ExecIf updates

/ExecIf

```
:%s/\(ExecIf(\)\( \$[.*]\),\(.*)\),\(.*)\)$\1\2?\3(\4)/
```

```
sed -e 's/\(ExecIf(\)\( \$[.*]\),\(.*)\),\(.*)\)$\1\2?\3(\4)/'
```

Deprecated stuff - dialplan

```
grep -ir \
```

```
-e SetCallerPres \
```

```
-e WaitMusicOnHold \
```

```
-e SetMusicOnHold \
```

```
-e QUEUE_MEMBER_COUNT \
```

```
-e Local \
```

```
*
```

Faxing

Changes that can be made on the Brother MFC-8860DN to better optimize for using with VOIP.

Resolution press MENU/SET, 2, 2, 2, choose standard. Press MENU/SET, STOP,EXIT.

Baud (baud rate)/ECM press MENU/SET, 2, 0, 1, which is compatiability. Press up or down arrow to choose basic. Press MENU/SET, STOP/EXIT. This sets baud to 9600 and disables ECM.

Building

```
cd /usr/src/asterisk/mpg123-0.59r-gpl  
make linux-x86_64 ; make install ; cd ..
```

```
cd /usr/src/asterisk/1.6.2.4
```

```
cd spandsp-0.0.6  
./configure ; make install ; ldconfig ; cd ..
```

```
cd corosync-1.2.0  
./configure ; make install ; cd ..
```

```
cd openais-1.1.2  
./configure ; make install ; cd ..
```

```
cd dahdi-linux-2.2.1  
make install ; cd ..
```

```
cd dahdi-tools-2.2.1  
./configure ; make menuconfig ; make install ; make config ; make samples ; cd ..
```

```
cd libpri-1.4.10.2  
make install ; cd ..
```

```
cd asterisk-1.6.2.4  
./configure ; make menuconfig ; make install ; make samples ; make config ; cd ..
```

```
cd asterisk-addons-1.6.2.0  
./configure ; make menuconfig ; make install ; make samples ; cd ..
```

```
cd ..
```

Simplify Asterisk 1.6 Dialplan Extensions

```
:%s/exten => .*,n/same => n/g
```

Sipura Distinctive Ring

```
exten => s,n,SIPAddHeader(Alert-Info: info=<Bellcore-r2>)
```

Available Distinctive Ring Patterns:

Pattern Name	Distinctive Ring Patterns
=====	=====
Bellcore-r1	60(2/4)
Bellcore-r2	60(.8/.4,.8/4)
Bellcore-r3	60(.4/.2,.4/.2,.8/4)
Bellcore-r4	60(.3/.2,1/.2,.3/4)
Bellcore-r5	1(.5/.5)
Bellcore-r6	60(.2/.4,.2/.4,.2/4)
Bellcore-r7	60(.4/.2,.4/.2,.4/4)
Bellcore-r8	60(0.25/9.75)

Available Distinctive Call Waiting Tone Patterns:

Pattern Name	Tone Pattern
=====	=====
Bellcore-r1	30(.3/9.7)
Bellcore-r2	30(.1/.1, .1/9.7)
Bellcore-r3	30(.1/.1, .1/.1, .1/9.7)
Bellcore-r4	30(.1/.1,.3/.1,.1/9.3)
Bellcore-r5	1(.5/.5)
Bellcore-r6	30(.1/.1,.3/.2,.3/9.1)
Bellcore-r7	30(.3/.1,.3/.1,.1/9.1)
Bellcore-r8	2.3(.3/2)

Using AWK to find individual calls

The below scripting will find all calls to the extension 11175 and print all the details for each call.

```
grep Dial /var/log/asterisk/full | grep macro-stdexten-v3 | grep "11175," | awk -F\['{print $3}' | awk -F\['{print $1}' | while read x; do echo ""; echo ""; echo ""; echo "identifier: $x"; echo ""; echo ""; echo ""; grep "\[$x\]" /var/log/asterisk/full; done
```

```
grep "Called .*11175" /var/log/asterisk/full | awk -F\['{print $3}' | awk -F\['{print $1}' | while read x; do echo ""; echo ""; echo ""; echo "identifier: $x"; echo ""; echo ""; echo ""; grep "\[$x\]" /var/log/asterisk/full; done
```

```
grep "Called .*11175" /var/log/asterisk/full | awk -F\['{print $3}' | awk -F\['{print $1}' | while read x; do echo ""; echo ""; echo ""; echo "identifier: $x"; echo ""; echo ""; echo ""; grep "\[$x\]" /var/log/asterisk/full; done | grep -A 5 -e "Called .*11175"
```

Finding hung channels

Asterisk 1.4,1.6

```
asterisk -rx "core show channels concise" | awk -F ! '{print $1,$11}'
```

```
asterisk -rx "core show channels concise" | awk -F ! '{print $1,$11}' | while read x y; do if [ $y -gt 60 ] || [ $y -lt 0 ]; then echo "$x - `expr $y / 60` minutes `expr $y % 60` seconds"; fi ; done
```

Asterisk 1.2

```
asterisk -rx "show channels concise" | awk -F : '{print $1,$11}'
```

Asterisk and SystemD

[See here...](#)

Put the following in /etc/systemd/system/asterisk.service and then run "systemctl daemon-reload && systemctl enable asterisk"

```
[Unit]
Description=Asterisk PBX And Telephony Daemon
Wants=network.target
After=network.target

[Service]
Type=simple
User=root
Group=root
#Environment=HOME=/var/lib/asterisk
#WorkingDirectory=/var/lib/asterisk
ExecStart=/usr/sbin/asterisk -f -C /etc/asterisk/asterisk.conf
ExecStop=/usr/sbin/asterisk -rx 'core stop now'
ExecReload=/usr/sbin/asterisk -rx 'core reload'

LimitNOFILE=65535

# safe_asterisk emulation
Restart=always
RestartSec=10

[Install]
WantedBy=multi-user.target
```

Paste below into a terminal to setup the files:

```
cat << EOF > /etc/systemd/system/asterisk.service
[Unit]
Description=Asterisk PBX And Telephony Daemon
Wants=network.target
After=network.target

[Service]
Type=simple
User=root
Group=root
#Environment=HOME=/var/lib/asterisk
```

```
#WorkingDirectory=/var/lib/asterisk
ExecStart=/usr/sbin/asterisk -f -C /etc/asterisk/asterisk.conf
ExecStop=/usr/sbin/asterisk -rx 'core stop now'
ExecReload=/usr/sbin/asterisk -rx 'core reload'

LimitNOFILE=65535

# safe_asterisk emulation
Restart=always
RestartSec=10

[Install]
WantedBy=multi-user.target

EOF
```

Loading Dahdi

After some digging, for my purposes I found the best way to load the transcoding module is to use the systemd module loader to load the wctc4xxp transcoding module and UDEV to run dahdi_cfg once the module is loaded.

Prerequisite: Configure the /etc/dahdi/modules and /etc/dahdi/system.conf files as you normally would.

```
cat << EOF > /etc/dahdi/modules
# /etc/modules-load.d/dahdi is symlinked here so systemd will load it on startup
# /etc/udev/rules.d/dahdi-wctc4xxp.rules instructs udev to run dahdi_cfg after wctc4xxp module is loaded

# Digium TC400B: G729 / G723 Transcoding Engine
wctc4xxp

EOF

ln -s /etc/dahdi/modules /etc/modules-load.d/dahdi.conf

# update udev to run dahdi_cfg after loading the transcoding module
cat << EOF > /etc/udev/rules.d/dahdi-wctc4xxp.rules
KERNEL=="wctc4xxp" RUN+= "/usr/sbin/dahdi_cfg"
EOF
```


The next time you reboot, systemd will load the module, udev will run dahdi_cfg, and then systemd will load asterisk. Granted this is only really needed if you haven't migrated away from MeetMe yet...

GROUP and GROUP_COUNT

func_GROUP

func_GROUP_COUNT

```
Set(GROUP(category)=groupname)
Set(CHECKING=${GROUP_COUNT(groupname@category)})
```

Example:

```
same => n,Set(GROUP(rcf)=${CHANNEL(accountcode)})
same => n,Set(GROUP(rcf${RCF_DID})=${CHANNEL(accountcode)})

same => n,Set(rcfCountAccountcode=${GROUP_COUNT(${CHANNEL(accountcode)})@rcf})
same => n,Set(rcfCountDID=${GROUP_COUNT(${CHANNEL(accountcode)})@rcf${RCF_DID}})
```

Asterisk comparisons

```
same => n,ExecIf("${testVariable}" = "SomeValue"?Set(testvar=TRUE):Set(testvar=FALSE))

; Numerical comparisons require not using quotes
same => n,Set(count=5)
same => n,Gofolf("${count} < 10"?trueTarget:falseTarget)

; Check if a variable is NOT NULL - This one will set testVariable2=TRUE
same => n,Set(testVariable1=something)
same => n,ExecIf("${LEN(testVariable1)}"?Set(testVariable2=TRUE):Set(testVariable2=FALSE))

; Check if a variable is NOT NULL - This one will set testVariable2=FALSE
same => n,Set(testVariable1=)
```

```
same => n,ExecIf(!$${LEN(${testVariable1}})]?Set(testVariable2=TRUE):Set(testVariable2=FALSE))
```

```
; Multiple comparisons using AND (&)
```

```
same => n,ExecIf("${testVariable1}" = "VALID" & "${testVariable2}" = "VALID"?set(status=TRUE))
```

```
; Multiple comparisons using OR (|)
```

```
same => n,ExecIf("${testVariable1}" = "VALID" | "${testVariable2}" = "VALID"?set(status=TRUE))
```

```
; Providing only a false conditional path
```

```
same => n,Set(testVariable1=1)
```

```
same => n,GotoIf(${ }?:falseCondition)
```

```
same => n(trueCondition),NoOp(do something on true condition)
```

```
same => n,Hangup()
```

```
same => n(falseCondition),NoOp(do something on false condition)
```

```
same => n,Hangup()
```

[end]

Revision #8

Created 6 August 2018 03:18:03 by bluecrow76

Updated 1 August 2022 01:13:26 by bluecrow76