

# TRUCS ASTERISK

*Mamadou Alpha BARRY*  
*Enseignant-Formateur ESMT*  
*BP :10 000 Liberté Dakar*  
*M : +221 77 403 0579*  
*F : +221 33 868 0311*  
*email : [alpha.barry@esmt.sn](mailto:alpha.barry@esmt.sn)*

## 1.

crée une variable dans le [globals]

```
[globals]
CHAIN=SIP/standard&SIP/toto1&SIP/toto2&SIP/toto3&SIP/toto4&SIP/toto5&SIP/to
to6&SIP/toto7&SIP/toto8&SIP/t...
```

puis : Code:

```
exten => 120,3,Dial(SIP/${CHAIN},30)
```

## 2.

```
#####
# Après une installation de Debian Lenny, avec "Base" choisi à
l'installation.
#####

# pour plus de confort, travaillez en ssh
apt-get install openssh-server

# téléchargement des archives
cd /usr/src
wget -c http://downloads.asterisk.org/pub/telephony/asterisk/asterisk-1.8-
current.tar.gz
wget -c http://downloads.asterisk.org/pub/telephony/dahdi-linux/dahdi-
linux-current.tar.gz
wget -c http://downloads.asterisk.org/pub/telephony/dahdi-tools/dahdi-
tools-current.tar.gz
wget -c http://downloads.asterisk.org/pub/telephony/libpri/libpri-1.4-
current.tar.gz

# dépaquetage
tar xvzf asterisk-1.8-current.tar.gz
tar xvzf dahdi-linux-current.tar.gz
tar xvzf dahdi-tools-current.tar.gz
tar xvzf libpri-1.4-current.tar.gz

# nécessaire pour toutes les compilations
apt-get install --assume-yes build-essential

# nécessaire pour dahdi
```

```

apt-get -y install linux-headers-$(uname -r)
ln -s linux-headers-$(uname -r) linux

# installation dahdi-linux
pushd /usr/src
cd dahdi-linux-*
make
make install
popd

# l'un ou l'autre est nécessaire pour afficher l'interface du make
menuselect
apt-get install -y \
libncurses-dev \
libnewt-dev

# installation dahdi-tools
pushd /usr/src
cd dahdi-tools-*
./configure
make menuselect
make
make install
make config
popd

# libss7
pushd /usr/src
wget http://downloads.digium.com/pub/telephony/libss7/libss7-1.0.2.tar.gz
tar -xvzf libss7-1.0.2.tar.gz
cd libss7-1.0.2
make && make install
popd

# installation libpri
pushd /usr/src
cd libpri-1.4.*
make
make install
popd

# dépendances nécessaires pour asterisk
apt-get install -y \
libz-dev \
libssl-dev \
libxml2-dev

# ou bien, si vous souhaitez débloquent un maximum d'options dans Asterisk
pushd /usr/src
cd asterisk-1.8.*
./contrib/scripts/install_prereq install
./contrib/scripts/install_prereq install-unpackaged
popd

# Channel Drivers > misdn
pushd /usr/src
wget -c http://www.misdn.org/downloads/releases/mISDN-1_1_9_2.tar.gz

```

```

wget -c http://www.misdn.org/downloads/releases/mISDNuser-1_1_9.1.tar.gz
tar xvzf mISDN-1_1_9_2.tar.gz
tar xvzf mISDNuser-1_1_9.1.tar.gz
popd

pushd /usr/src
cd mISDN-1_1_9_2
make && make install
popd

apt-get install -y flex
pushd /usr/src
cd mISDNuser-1_1_9.1
make && make install
popd

# Resource Modules > res_srtp
pushd /usr/src
wget -O srtp-1.4.4.tgz
http://sourceforge.net/projects/srtp/files/srtp/1.4.4/srtp-1.4.4.tgz/download
tar xvzf srtp-1.4.4.tgz
cd srtp
./configure && make && make install
# le make runtest de la 1.4.4 est buggue
popd

# Voicemail Build Options > IMAP_STORAGE
apt-get install -y uw-mailutils libc-client2007b-dev

# Compiler Flags > REBUILD_PARSERS
apt-get install -y bison

# Resource Modules > res_calendar_caldav res_calendar_exchange
res_calendar_icalendar
pushd /usr/src
wget -O libical-0.44.tar.gz
http://sourceforge.net/projects/freeassociation/files/libical/libical-0.44/libical-0.44.tar.gz/download
# il existe une version 0.46, mais un bug empeche pour l'instant sa
compilation:
http://sourceforge.net/tracker/index.php?func=detail&aid=3072673&group_id=16077&atid=116077
tar zxvf libical-0.44.tar.gz
cd libical-0.44
./configure --enable-shared && make && make install
popd

# Resource Modules > res_calendar_ews
pushd /usr/src
wget -c http://www.webdav.org/neon/neon-0.29.5.tar.gz
tar xvzf neon-0.29.5.tar.gz
cd neon-0.29.5
./configure --enable-shared --with-ssl && make && make install
popd

# app_osplookup
pushd /usr/src
wget -O OSPToolkit-3.6.1.tar.gz http://sourceforge.net/projects/osp-toolkit/files/osp-toolkit/OSPToolkit%203.6.1/OSPToolkit-3.6.1.tar.gz/download

```

```

tar xvzf OSPToolkit-3.6.1.tar.gz
cd TK-3_6_1-20100107/src
make clean && make build && make install
# et pour asterisk: ./configure --with-osptk=/usr/local/include/osp
popd

# hoard - Use the Hoard Memory Allocator instead of the default system one
pushd /usr/src
wget -c http://www.cs.umass.edu/%7Eemery/hoard/hoard-3.8/source/hoard-38.tar.gz
tar xvzf hoard-38.tar.gz
cd hoard-38/src
make linux-gcc-x86
# et pour asterisk: ./configure --with-hoard=/usr/src/hoard-38/src
popd

pushd /usr/src
cd asterisk-1.8.*
contrib/scripts/get_mp3_source.sh
contrib/scripts/get_ilbc_source.sh
./configure --with-osptk=/usr/local/include/osp --with-hoard=/usr/src/hoard-38/src
make menuselect
make
make install
make samples
make config
popd

# si utilisation de misdn
mISDN scan
mISDN config
misdn-init scan
misdn-init config
misdn-init start
mISDN start

```

### 3.

**Il arrive fréquemment que le numéro d'appelant ne soit pas présenté dans le bon format à votre Asterisk.**

**En effet selon le type de ligne, l'opérateur, la provenance de l'appel, le callerid peut être présenté sous différentes formes: 10 chiffres, 9 chiffres, un zéro de trop, préfixe international malgré que l'appel vienne du même pays.**

**Ces différences peuvent être gênantes: difficulté de recherche dans les journaux d'appels, multiplication des lignes de dialplan pour gérer les différents formats, etc...**

**Ce morceau de dialplan permet de uniformiser le format du callerid:**

#### **extensions.conf**

```

[appels-entrants]
exten => _X.,1,noop(appel entrant)
same => n,Gosub(correction-callerid,1,1)
same => n,Dial(SIP/.....)
same => n,hangup()

```

```
[correction-callerid]
exten => 1,1,noop(Correction du format du callerid)

; callerid vide
same => n,GotoIf(${${CALLERID(num)} = ""}?fincorrection)

; callerid a neuf chiffres (ex: 155779988, le zero est manquant)
same => n,GotoIf(${${LEN(${CALLERID(num)})} = 9}?neufchiffres)

; callerid avec le prefixe international (ex: 33155779988)
same => n,GotoIf(${${CALLERID(num):0:2} = 33}?debut33)

; callerid avec un zero en trop
same => n,GotoIf(${${${LEN(${CALLERID(num)})} = 11 ] &
${${CALLERID(num):0:2} = 00}}?onzechiffresdeuxzero)

; callerid deja au bon format, ou autre cas non-pris en charge
same => n,goto(fincorrection)

; les corrections associees
same => n(neufchiffres),Set(CALLERID(num)=0${CALLERID(num)})
same => n(neufchiffres),goto(fincorrection)
same => n(debut33),Set(CALLERID(num)=0${CALLERID(num):2})
same => n(debut33),goto(fincorrection)
same => n(onzechiffresdeuxzero),Set(CALLERID(num)=${CALLERID(num):1})
same => n(onzechiffresdeuxzero),goto(fincorrection)

; affichage du callerid corrige dans la CLI
same => n(fincorrection),NoOP(CALLERID->${CALLERID(num)})
same => n,Return()
```

#### 4.

## (Petit) Lexique

**PABX** (*Private Automatic Branch Exchange*) c'est le nom utilisé pour designer un autocommutateur ou central téléphonique, aujourd'hui, notre PABX à nous s'appelle **ASTERISK** : c'est un PABX "logiciel" (IPBX) contrairement à beaucoup d'autres qui sont matériel, et très couteux, Asterisk est OpenSource, et gratuit. (<http://www.asterisk.org>)

**IPBX** : PABX ip

**FXS** (*Foreign eXchange Subscriber*) : c'est le nom d'une carte installée dans votre PC, qui vous permettra de convertir un signal entrant numerique, en signal analogique pour y brancher votre téléphone "analogique" derrière.

**FXO** (*Foreign eXchange Office*) : c'est le nom de la carte, sur laquelle vous branchez votre ligne téléphonique, cette carte sera chargée de recevoir les appels entrant "analogiques" et les convertir en signaux numeriques.

**PSTN** (*Public Switched Telephone Network*) : c'est le nom d'un fournisseur téléphonique, Wengo, Directcentrex, Phonesystems, sont des PSTN.

**SIP et IAX2** : Voir : PROTOCOLE.

[\[modifier\]](#)

## Introduction

Vous avez besoin de créer un réseau téléphonique entre un groupe de personnes, créer des numéros internes abrégés, disposer de votre propre musique d'attente, transférer des appels... ou bien faire un système de serveur vocal couplé à une base MySQL, avec des menus, une identification ou l'enregistrement de messages vocaux pour vos sites internet ? Alors, vous avez besoin d'installer ASTERISK.

Je vais tenter de vous expliquer l'installation et le début de la configuration , mais les possibilités sont énormes, et il me serait impossible de rentrer dans les détails de chaque option.

Pour commencer,

ASTERISK est un logiciel serveur OpenSource PBX, c'est lui, qui va gérer votre identification au serveur, la réception et l'émission des appels par internet.

Pour réaliser ce didacticiel/howto, j'ai utilisé un serveur dédié (testé sur : Redhat, Gentoo, Mac osX), et un abonnement chez le fournisseur voIP : [Wengo](#) (du groupe 9 telecom). L'abonnement Wengo inclut 2 lignes (in ET out). L'inscription est rapide, et vous avez votre numéro en quelques minutes.

Il existe plusieurs autres opérateurs comme : [Directcentrex](#) ou [Phonesystems](#) , et [Phonext](#) pour la Belgique. Un fournisseur de lignes entrantes (pas d'appel sortant) propose des numéros de téléphone publics à bas prix, notamment en Belgique, il s'agit de [VoxBone](#). Je ne les ai pas testés, mais ils sont tous très comparables, et leur fonctionnement est identique, vous pouvez donc choisir celui qui vous conviendra le mieux.

[\[modifier\]](#)

## En 10 minutes (chrono!) ...

Pour ceux qui n'ont pas le temps....

```
wget http://ftp.digium.com/pub/asterisk/asterisk-1.2.7.1.tar.gz
wget http://ftp.digium.com/pub/asterisk/asterisk-addons-1.2.1.tar.gz
wget http://ovh.dl.sourceforge.net/sourceforge/sox/sox-12.17.8.tar.gz
wget http://www.sineapps.com/FrenchPrompts.tar.gz
wget http://www.mpg123.de/mpg123/mpg123-0.59r.tar.gz

tar -xzf asterisk-1.2.7.1.tar.gz
tar -xzf asterisk-addons-1.2.1.tar.gz
tar -xzf sox-12.17.8.tar.gz
tar -xzf FrenchPrompts.tar.gz
tar -xzf mpg123-0.59r.tar.gz
```

```
cd mpg123-0.59r
make linux
make install

cd ../sox-12.17.8
./configure
make
make install

cd ../asterisk-1.2.7.1
make
make install
make samples

cd ../asterisk-addons-1.2.1/
make
make install

cd ../
mv fr /var/lib/asterisk/sounds/.
mv digits/fr /var/lib/asterisk/sounds/digits/fr
\[modifier\]
```

## Protocole

Il existe plusieurs protocoles d'identification pour la voIP, le standard aujourd'hui s'appelle "SIP" (Session Initiation Protocol), c'est un standard ouvert, le serveur Asterisk a son propre protocole "IAX2", mais asterisk sait très bien gérer les deux protocoles, et ceux-ci sont même compatibles entre eux. (exemple : un softPhone en IAX2, un fournisseur voIP en IAX2, et un softphone en SIP : tous branchés à Asterisk !)

Un fournisseur voIP va donc vous fournir, non pas un fil arrivant chez vous pour brancher votre téléphone, mais simplement quelques paramètres : l'adresse IP de leur serveur, un login, et un mot de passe ! Cela suffit aujourd'hui pour téléphoner depuis n'importe où !

[\[modifier\]](#)

## Téléchargement

Ici, nous allons télécharger l'ensemble des applications utiles pour l'installation d'Asterisk. Les versions vont surement évoluer, méfiez vous des liens, et allez éventuellement télécharger directement les applications sur leur site respectif pour prendre la dernière version.

Le logiciel "Asterisk"

```
# wget http://ftp.digium.com/pub/asterisk/asterisk-1.2.7.1.tar.gz
```

les Addons d'Asterisk, vont permettre l'utilisation d'Asterisk avec MySQL :

```
# wget http://ftp.digium.com/pub/asterisk/asterisk-addons-1.2.1.tar.gz
```

Le logiciel SOX vous permet de convertir un fichier "WAV" en format "GSM" qui est le format pour faire diffuser une annonce d'accueil par exemple.

```
# wget http://ovh.dl.sourceforge.net/sourceforge/sox/sox-12.17.8.tar.gz
```

Ceci est un package de voix en français, plutôt que la version standard en anglais (utilisée pour le repondeur, par exemple) les voix ne sont pas de très bonne qualité, mais ont l'avantage tout de même d'être en français.

```
# wget http://www.sineapps.com/FrenchPrompts.tar.gz
```

mpg123 est un logiciel permettant de diffuser un contenu en MP3, il est indispensable pour utiliser la musique d'attente.

```
# wget http://www.mpg123.de/mpg123/mpg123-0.59r.tar.gz  
[modifier]
```

## Installation

On décompresse maintenant l'ensemble des logiciels téléchargés

```
# tar -xzf asterisk-1.2.7.1.tar.gz  
# tar -xzf asterisk-addons-1.2.1.tar.gz  
# tar -xzf sox-12.17.8.tar.gz  
# tar -xzf FrenchPrompts.tar.gz  
# tar -xzf mpg123-0.59r.tar.gz
```

On va commencer par installer sox et mpg123, qui sont des petits logiciels assez rapide à installer.

```
# cd mpg123-0.59r  
# make linux  
# make install
```

On installe ensuite SOX

```
# cd ../sox-12.17.8  
# ./configure  
# make  
# make install
```

On installe maintenant le serveur Asterisk

```
# cd ../asterisk-1.2.7.1  
# make  
# make install  
# make samples
```



make samples va installer les fichiers de configuration par default. (Lors de vos mises a jours, plus la peine de l'executer)

On installe maintenant les addons Mysql

```
# cd ../asterisk-addons-1.2.1/  
# make  
# make install
```

On installe les voix francaises

```
# cd ../  
# mv fr /var/lib/asterisk/sounds/.  
# mv digits/fr /var/lib/asterisk/sounds/digits/fr
```

[\[modifier\]](#)

## Configuration

Voila, votre serveur est installé, il faut maintenant faire la "configuration". L'ensemble de vos fichiers de configuration se trouve dans le dossier /etc/asterisk/

```
# cd /etc/asterisk/
```

Seuls quelques fichiers nous interessent, à savoir :

```
sip.conf  
extensions.conf
```

Le fichier sip.conf, va définir l'ensemble de vos utilisateurs, le fichier extensions.conf, va définir les actions à faire, lors d'un appel sur un numéro de téléphone. Pour connaitre l'ensemble des fonctions utilisables, rendez-vous sur le site <http://www.asteriskguru.com/tutorials/> et également sur le site <http://www.voip-info.org/>

Editez le fichier sip.conf, à la fin du fichier et créez vos utilisateurs ainsi :

```
[jerome]  
type=friend  
username=jerome  
secret=MonPaSsw0rd  
host=dynamic  
context=moncontexte  
language=fr  
insecure=very  
nat=yes  
canreinvite=no
```

```
[francois]  
type=friend
```

```
username=francois
secret=MonPaSsw0rd
host=dynamic
context=moncontexte
language=fr
insecure=very
nat=yes
canreinvite=no
```

Ici, nous avons créé 2 utilisateurs : jerome et francois, vous pouvez bien sur en créer autant que vous voulez.

```
type=friend : va définir que votre utilisateur est un "ami" est il sera
autorisé à passer des appels SORTANT
host=dynamic : permet de se connecter avec cet utilisateur depuis n'importe
quelle IP.
context=moncontexte : l'option "context" sera ensuite utilisée dans le
fichier extensions.conf, lorsque l'utilisateur emit un appel, il rentrera
dans la zone "moncontexte", qu'on definira plus tard.
language=fr : permet d'utiliser les fichiers audio francais.
```

Nous allons ensuite configurer notre compte WENGO

Rendez-vous sur : [http://ws.wengo.fr/softphone-sso/sso.php?login=VOTRE\\_EMAIL&password=VOTRE\\_PASS](http://ws.wengo.fr/softphone-sso/sso.php?login=VOTRE_EMAIL&password=VOTRE_PASS) en modifiant VOTRE\_EMAIL et VOTRE\_PASS par vos identifiants Wengo.

Ouvrez ensuite la source de ce fichier, vous y trouverez un flux XML :

```
<?xml version="1.0" ?>
- <#> <sso vapi="*1*">
<status code="*200*">OK</status>
<user>ident-wengo</user>
<userid>ident-wengo</userid>
<displayname>ident-wengo</displayname>
<password>XXXXXXXXXXXXXXXXXX</password>
<outbound>1</outbound>
<realm>voip.wengo.fr</realm>
- <#> <proxy>
<host>213.91.9.210</host>
<port>5060</port>
</proxy>
- <#> <server>
<host>voip.wengo.fr</host>
<port>5060</port>
</server>
- <#> <httptunnel>
<host>httpproxy.wengo.fr</host>
</httptunnel>
</sso>
```

Recuperez votre "ident-wengo" et votre mot de passe "XXXXXXXXXXXXXXXXXX" : il est

different de votre compte Wengo, ceci est votre mot de passe "SIP", cette étape est indispensable pour connaître ce parametre.

Une fois ces elements en main, éditez à nouveau le fichier sip.conf, et rajoutez dans la section [general], la ligne suivante :

```
register => ident-wengo:XXXXXXXXXXXXXXXXXX@wengo/ident-wengo
```

Ensuite, nous allons définir l'utilisateur @"wengo", tout comme vos utilisateurs (ex : jerome, francois...) à la difference qu'au lieu d'être "type=friends" c'est "type=peer"

On met donc à la fin du fichier sip.conf :

```
[wengo]
type=peer
fromdomain=voip.wengo.fr
host=voip.wengo.fr
outboundproxy=sipproxy.wengo.fr
outboundproxyport=5060
username=ident-wengo
fromuser=ident-wengo
secret=XXXXXXXXXXXXXXXXXX
insecure=very
nat=yes
canreinvite=no
```

Toujours dans la section [general] vous allez devoir modifier les parametres suivants, afin d'assurer le bon fonctionnement :

Modifiez chacune des lignes suivante :

```
context=default
;allowguest=no
;dtmfmode = rfc2833
;notifyringing = yes
;relaxdtmf=yes
```

en :

```
context=wengo-in
allowguest=no
dtmfmode = auto
notifyringing = yes
relaxdtmf=yes
```

allowguest, autorise la connexion au serveur ASTERISK à n'importe quelle personne : afin d'éviter ça, on bloque donc les utilisateurs.

le context dans la section [general] décrit le contexte dans lequel un appel arrive et ce qu'on doit en faire... on l'appelle "wengo-in" !

Fermer le fichier SIP : normalement, c'est tout bon :-)

Maintenant, nous allons configurer le fichier extensions.conf :

Rendez vous a la fin du fichier, nous allons d'abord créer le context [wengo-in]

```
[wengo-in]
exten => _.,1,Goto(moncontexte,100,1)
```

Lorsqu'un appel arrivera sur wengo, le serveur Asterisk ira immédiatement dans le contexte : "moncontexte", au numero 100, de la priorité 1.

Nous allons maintenant créer la zone "moncontexte"

```
[moncontexte]

exten => 100,1,Ringing
exten => 100,2,Wait,1
exten => 100,3,Answer
exten => 100,4,Dial(SIP/jerome&SIP/francois,50,tm)
exten => 100,5,Hangup

exten => 101,1,Dial(SIP/jerome,20,tm)
exten => 101,2,Wait,5
exten => 101,3,Dial(SIP/francois,20,tm)
exten => 101,4,Hangup

exten => 102,1,Dial(SIP/francois,20,tm)
exten => 102,2,Wait,5
exten => 102,3,Dial(SIP/jerome,20,tm)
exten => 102,4,Hangup

; -----
; Pour passer un appel (en france)
exten => _0[123456].,1,Dial(SIP/wengo/${EXTEN},240,T)
exten => _0[123456].,2,Hangup

; Numero GRATUIT
exten => _080[05].,1,Dial(SIP/wengo/${EXTEN},240,T)
exten => _080[05].,2,Hangup

; Numero INTERNET
exten => _087.,1,Dial(SIP/wengo/${EXTEN},240,T)
exten => _087.,2,Hangup
; -----
```

Cette zone moncontexte, définit plusieurs choses :

Dans un premier temps, lorsqu'un appel arrive sur Wengo, on le redirige vers le numero interne 100, priorité 1.

On laisse sonner, puis, à la priorité 2, on attend, priorité 3, on décroche, priorité 4 : On fais sonner les 2 numeros de telephone "jerome" et "francois", qui sont nos 2 comptes SIP.

Que signifie 50 ?

50 est le timeout, en cas de non-reponse au bout de 50 secondes, nous passerons automatiquement à la priorité 5 : Hangup (on raccroche)

Que signifie "tm" ?

t, signifie que la personne appelée pourra transférer un appel.  
T, signifie que la personne qui appelle pourra transférer un appel.

m, l'option "m" correspond à la musique d'attente : lorsque vous executerez un "transfert" d'appel, votre correspondant entendra vos musiques d'attente situées dans le dossier : /var/lib/asterisk/mohmp3

Le numero interne 101 est, par contre, uniquement pour l'utilisateur "jerome".

Si l'utilisateur "francois" prend l'appel entrant, mais que l'appel est pour "jerome", il suffira alors de taper les touches #101 pour transferer immédiatement l'appel à Jerome. En cas de non reponse dans les 20 secondes, l'appel reviendra sur francois.

Et inversement pour le numero 102.

Dans l'écriture d'un numéro de téléphone, vous pouvez commencer votre numéro par "\_". Ce signe signifie que votre numéro de téléphone sera écrit sous forme de "pattern". Vous pourrez ainsi utiliser [12345] pour un groupe de numéros, le signe "." pour définir une chaine illimitée, X pour définir un chiffre de 0 a 9.

[\[modifier\]](#)

## Démarrage du serveur

Pour démarrer automatiquement votre serveur Asterisk au démarrage de la machine, vous devez enregistrer la commande au démarrage de votre serveur (a l'aide de WebMin ?) :

```
/usr/sbin/safe_asterisk
```

Vous pouvez également demarrer depuis la console, en tapant la commande :

```
asterisk -vvvc
```

Une fois le serveur "demarré", vous pouvez vous connecter à la console en tapant la commande :

```
asterisk -r
```

Cette commande ne demarre pas le serveur, elle vous permet juste de vous reconnecter sur la console une fois le serveur deja lancé !

Note : si vous n'avez pas postgresql d'installé sur le serveur, reportez vous un peu plus bas pour désactiver le module se référant à ce serveur de base de données.

[\[modifier\]](#)

## Commandes du serveur

Pour connaître l'ensemble des commandes du serveur, vous pouvez taper : help, une fois connecté à la console Asterix

Ceci dit, voici quelques commandes utiles :

A chaque modification du fichier sip.conf, vous devez executer la commande "sip reload" pour recharger le fichier

A chaque modification du fichier extensions.conf, vous devez executer la commande "extensions reload" pour recharger le fichier.

Pour recharger l'ensemble du serveur, tapez la commande : reload

Pour connaître l'ensemble des "peers" connectés (vos utilisateurs SIP), tapez : sip show peers

Pour vérifier que votre connexion avec Wengo est bien "enregistrée", tapez : sip show registry

[\[modifier\]](#)

## Softphone ou Hardphone ?

Voilà, nous avons fini d'installer et de configurer le serveur, maintenant il vous faut un telephone ! Le choix est à vous : pour commencer simplement, avec un Softphone, téléchargez le logiciel X-LITE [Xten](#), puis allez dans :

Preferences > System Settings > SIP PROXY > [Default] >

Puis, configurez ainsi :

```
Enabled : YES (ne pas oublier de changer!!!)
Display Name : jerome
Username : jerome
Authorization User : jerome
Password : MonPasSword
Domain/relam: IPduServeur
SIP Proxy : IPduServeur
Out Bound Proxy : IPduServeur
```

fermez les préférences.... c'est prêt !

Note: avec certains firewall il est bon de configurer SIP Proxy en précisant le port utilisé (par défaut 5060, sinon vous trouverez la valeur dans sip.conf (ligne dans sip.conf : bindport=5060 (Pour X-Lite ca donne : IPduServeur:5060 )

[\[modifier\]](#)

## Un peu plus loin... (en vrac!)

### SOX

Pour utiliser SOX, et convertir un fichier wav (ou autre comme mp3) en format GSM, utilisez la commande suivante :

```
sox foo.wav -r 8000 foo.gsm resample -ql
```

ou bien

```
sox foo.mp3 -r 8000 foo.gsm resample -ql
```

**Attention cependant :)))** si vous utilisez un fichier source qui est stéréo (*nombre de canaux supérieur à 1*), vous devez lui demander de le convertir en mono, sinon vous risquer d'entendre un son *très bizarre...*

```
sox foo.mp3 -r 8000 -c1 foo.gsm resample -ql
```

### EXTEN

Pour faire écouter un message d'accueil avant de renvoyer vers une personne, vous devez utiliser la fonction : Playback.

Exemple :

```
exten => 100,1,Playback(bonjour)
```

le fichier doit se trouver dans votre dossier "sounds", en format .gsm. Dans l'appel de la fonction Playback, vous ne devez pas mettre l'extension .gsm

Maintenant, je vais vous montrer comment réaliser un message d'accueil en autorisant l'utilisateur à choisir vers quel poste il desire appeler.

En reprenant notre exemple d'installation avec [wengo-in] et [moncontexte] de tout a l'heure :

```
[wengo-in]
exten => _.,1,Goto(moncontexte,s,1)
```

```
[moncontexte]
```

```

; on laisse sonner
exten => s,1,Ringing
exten => s,2,Wait,2
; On décroche
exten => s,3,Answer
exten => s,4,Set(TIMEOUT(digit)=3)
exten => s,5,Set(TIMEOUT(response)=5)
; on diffuse le message de "bonjour"
exten => s,6,Background(msg-bonjour)
; on attend que l'utilisateur appuie sur une touche, puis, la touche
definira la nouvelle "extension" sur le quel l'utilisateur ira.
exten => s,7,WaitExten

; Si l'utilisateur appuie sur la touche 1, cela appellera le poste de
JEROME, si il appuie sur 2, il ira sur le poste de Francois.

; Poste interne "JEROME"
exten => 1,1,Dial(SIP/jerome,50,tm)
exten => 1,2,Hangup

; Poste interne "FRANCOIS"
exten => 2,1,Dial(SIP/francois,50,tm)
exten => 2,2,Hangup

; si l'utilisateur n'a pas appuyé sur une touche pendant le temps défini,
alors le systeme executera l'extension "t" (timeout) et on retourne à la
priorite 6 qui redifusera le message

exten => t,1,Goto(menu,s,6)

; Si l'utilisateur appuie sur une touche INVALIDE, le system executera
l'extension "i" automatiquement.

exten => i,1,Playback(desole) ; on met un message pour dire qu'on a pas
compris la touche...

exten => i,2,Goto(moncontexte,s,6) ; on le renvoie au menu

```

**ATTENTION** avec la fonction WaitExten. Evitez d'autoriser votre utilisateur à passer n'importe quel appel, les differents "contextes" limitent justement cette utilisation.

Si vous utilisez par exemple (pour passer un appel) :

```
exten => _0[123456].,1,Dial(SIP/wengo/${EXTEN},240,T)
```

Soyez donc vigilant dans la programmation de vos contextes.

Playback execute un fichier son, et l'utilisateur est obligé de l'entendre jusqu'à la fin.

Background, execute un fichier son egalement, mais avec la fonction WaitExten, il permet à l'utilisateur d'appuyer sur une touche, même pendant sa diffusion.



[\[modifier\]](#)

## PHP

Une autre fonction très utile : `System()`, elle vous permet d'exécuter des commandes système sur le serveur !

Exemple :

```
exten => _.,1,System(PHP /home/script.php ${EXTEN} ${CALLERIDNUM})
```

[\[modifier\]](#)

## MySQL

Maintenant, voici un exemple avec l'utilisation de MySQL :

```
exten => 555,1,MYSQL(Connect connid localhost USER PASS DATABASE)
exten => 555,2,MYSQL(Query resultid ${connid} SELECT\ id\,name\ FROM\
membres\ WHERE\ telephone=${CALLERIDNUM})
exten => 555,3,MYSQL(Fetch fetchid ${resultid} idm auteur)
exten => 555,4,MYSQL(Clear ${resultid})
exten => 555,5,MYSQL(Disconnect ${connid})
exten => 555,6,Noop(APPEL DE - ${idm} - ${auteur})
```

[\[modifier\]](#)

## Horloge parlante

Autre exemple ? Une horloge parlante :

Ca ne sert pas à grand chose, mais, voici comment faire une horloge parlante

```
exten => 777,1,Answer
exten => 777,2,SayUnixTime(,CET,kM)
exten => 777,3,Hangup
```

[\[modifier\]](#)

## Testez votre ligne ?

Vous pouvez, si vous désirez, faire un "test d'écho" : vous parlez, il vous retournera immédiatement ce qu'il reçoit.

Il faut utiliser, pour ça, la fonction : `Echo()`

```
exten => 200,1,Answer
exten => 200,2,Echo
exten => 200,3,Hangup
```

Appellez le numero : 200, parlez !

[\[modifier\]](#)

## Boites Vocale

L'installation d'une boite vocale est assez simple. Commencez par éditer le fichier /etc/asterisk/voicemail.conf

Modifiez les lignes suivantes :

```
format=wav
serveremail=votre@email.fr
attach=yes
fromstring=VOTRE SOCIETE
mailcmd=/usr/sbin/sendmail -t
```

Définissez ensuite votre contexte :

```
[default]
1000 => 9999,Jerome,votre@email.fr
```

1000 étant le numéro de ma boite vocale. (independant du numero de téléphone !)

9999 le mot de passe

Suivi de mon nom, et mon email sur le quel sera envoyé le fichier attaché.

Lors de la reception d'un appel, voila à quoi ressemblera notre fichier extensions.conf

```
exten => 101,1,Dial(SIP/jerome&SIP/francois,50,tm)
exten => 101,2,VoiceMail(1000@default)
exten => 101,3,Hangup
```

L'utilisateur ira, apres 50 secondes et automatiquement, sur le repondeur numero 1000, du contexte [default]

## Comment faire ensuite pour "consulter" ses messages ?

C'est très simple : vous devez créer un numéro sur lequel vous effectuez votre appel, et utiliser la fonction VoiceMailMain()

Exemple du fichier extensions.conf

```
; numero de la messagerie
exten => 222,1,Answer
```

```
exten => 222,2,VoiceMailMain(1000@default)
exten => 222,3,Hangup
```

Lors de l'appel au numero 222, vous tomberez directement sur la boite vocale 1000@default  
Le mot de passe vous sera demandé, et vous accederez ensuite à vos messages.

Certain telephone peuvent egalement vous fournir grace a un petit icone, la presence d'un ou plusieurs messages sur votre boite vocale. vous devez, pour cela, rajouter dans le contexte de votre utilisateur SIP la ligne :

```
mailbox=1000@default
```

Qui definira la "boite vocale" de cet utilisateur.

[\[modifier\]](#)

## **Fax**

Sachez également qu'il est possible de recevoir et d'envoyer des fax avec Asterisk, malheureusement, et même si j'ai pu avec beaucoup de mal installer les applications rxfax et txfax, je n'ai malheureusement pas réussi à obtenir un résultat suffisamment correct au niveau de la qualité de reception, surement du à la qualité de ligne Wengo. je n'expliquerais donc pas l'installation, relativement longue, tant que je n'aurais pas réussi à avoir un bon résultat.

Pour ceux qui veulent se lancer tout de même, il vous faudra installer :

- eps-1.5
- gcc > 3
- tiff-v3.6.1
- spandsp-0.0.2
- rxfax/txfax

[\[modifier\]](#)

## **Manager**

Asterisk vous permet de prendre son contrôle à distance, grâce à l'option "manager".

Il existe aujourd'hui quelques applications permettant de surveiller le serveur, ou envoyer des commandes, grâce au PHP, en utilisant les sockets par le port 5038.

Vous devez pour cela, créer un compte utilisateur au manager, et lui donner certains droits.

Ajoutez dans le fichier : `/etc/asterisk/manager.conf`

```
[jerome]
secret = MonPassword
read = system,call,log,verbose,command,agent,user ; all, peu suffir! ;)
write = system,call,log,verbose,command,agent,user
```

Rendez vous sur ce site <http://www.voip-info.org/wiki-Asterisk+manager+API>, pour connaître l'ensemble des fonctions liées au manager.

[\[modifier\]](#)

## Problemes ?

Si vous avez un problème au chargement de votre serveur Asterisk (problème avec un module ?)

Il se peut que vous soyez obligé de désactiver un module, pour X raisons.

Normalement, il ne devrait pas y avoir de problème, mais au cas ou, voici la methode pour lancer votre serveur en supprimant certain module qui ne fonctionnerait pas.

Vos modules se trouvent dans le dossier : `/usr/lib/asterisk/modules`

Editez le fichier : `/etc/asterisk/modules.conf`

Les modules sont automatiquement chargés, à partir du moment ou ils se trouvent dans ce dossier et si le parametre *autoload=yes* est bien definit à "yes"

Vous pouvez, cependant, désactiver les modules.

Dans le contexte `[modules]` de notre fichier `modules.conf`

Vous pouvez taper :

```
noload => cdr_pgsql.so
noload => codec_lpc10.so
```

La fonction *noload*, evitera le chargement de ces modules, vous pourrez ainsi démarrer votre serveur et le faire fonctionner, sans pouvoir utiliser les modules en question, evidement.

[\[modifier\]](#)

## F.A.Q.

**Que ce passe t'il lorsqu'un 2eme appel arrive sur le meme numéro ?**

Avec un abonnement comme wengo, vous disposer de plusieurs lignes in/out : 2, pour etre exact, avec phonesystems, ils vous est possible de prendre un abonnement avec autant de ligne que vous desirez. pour eviter que votre numéro soit occupé. les lignes sont completement separer, pas comme l'analogique, avec le "signal sonore", en numerique : si vous avez 2 lignes, 2 personnes differentes peuvent prendre 2 appels. ou 3, ou 4, en fonction de votre abonnement. Le choix de votre telephone sera lui aussi important : puisque c'est lui qui repondra a asterisk "OCCUPE", si il ne sais pas gerer plusieurs ligne differente.

le SoftPhone x-lite peu gerer jusqu'a 3 lignes.

le SoftPhone IDEFISK (protocole IAX2) , sais gerer lui jusqu'a 6 lignes differente.

### **res\_musiconhold.c:507 monmp3thread: Request to schedule in the past?!?!**

Cette erreur est du au manque d'un "timer" Sur votre serveur : rien de tres grave, elle provoquera cependant des petits sauts dans la lecture de la musique d'attente en mp3. le timer est habituellement sur les cartes de chez DIGIUM, vous pouvez cependant en emuler un, grace au package : Zaptel, en installant Zaptel/Ztdummy. (Bon courage.....) :-)

## **5.**

### **[Back to Tutorials](#)**

## **2. Asterisk 1.4.0 CLI commands**

### **Introduction**

Asterisk CLI supports large variety of commands which can be used for testing, configuration and monitoring. In this tutorial we will describe all commands available at the standard Asterisk version 1.4.0. We will divide this tutorial into few sections in order to facilitate the reading.

### **General CLI commands**

**!** - Execute a shell command

**abort halt** - Cancel a running halt

**cdr status** - Display the CDR status

**feature show** - Lists configured features

**feature show channels** - List status of feature channels

**file convert** - Convert audio file

**group show channels** - Display active channels with group(s)

**help** - Display help list, or specific help on a command

**indication add** - Add the given indication to the country

**indication remove** - Remove the given indication from the country

**indication show** - Display a list of all countries/indications

**keys init** - Initialize RSA key passcodes

**keys show** - Displays RSA key information

**local show channels** - List status of local channels

**logger mute** - Toggle logging output to a console  
**logger reload** - Reopens the log files  
**logger rotate** - Rotates and reopens the log files  
**logger show channels** - List configured log channels  
**meetme** - Execute a command on a conference or conferee  
**mixmonitor** - Execute a MixMonitor command.  
**moh reload** - Music On Hold  
**moh show classes** - List MOH classes  
**moh show files** - List MOH file-based classes  
**no debug channel (null)**  
**originate** - Originate a call  
**realtime load** - Used to print out RealTime variables.  
**realtime update** - Used to update RealTime variables.  
**restart gracefully** - Restart Asterisk gracefully  
**restart now** - Restart Asterisk immediately  
**restart when convenient** - Restart Asterisk at empty call volume  
**sla show** - Show status of Shared Line Appearances  
**soft hangup** - Request a hangup on a given channel  
**stop gracefully** - Gracefully shut down Asterisk  
**stop now** - Shut down Asterisk immediately  
**stop when convenient** - Shut down Asterisk at empty call volume  
**stun debug** - Enable STUN debugging  
**stun debug off** - Disable STUN debugging  
**udptl debug** - Enable UDPTL debugging  
**udptl debug ip** - Enable UDPTL debugging on IP  
**udptl debug off** - Disable UDPTL debugging

### **AEL commands**

**ael debug contexts** - Enable AEL contexts debug (does nothing)  
**ael debug macros** - Enable AEL macros debug (does nothing)  
**ael debug read** - Enable AEL read debug (does nothing)  
**ael debug tokens** - Enable AEL tokens debug (does nothing)  
**ael nodebug** - Disable AEL debug messages  
**ael reload** - Reload AEL configuration

### **Agents commands**

**agent logoff** - Sets an agent offline  
**agent show** - Show status of agents  
**agent show online** - Show all online agents

### **AGI commands**

**agi debug** - Enable AGI debugging  
**agi debug off** - Disable AGI debugging  
**agi dumphtml** - Dumps a list of agi commands in html format  
**agi show** - List AGI commands or specific help  
**dnsmgr reload** - Reloads the DNS manager configuration  
**dnsmgr status** - Display the DNS manager status  
**http show status** - Display HTTP server status

### Console commands

**console active** - Sets/displays active console  
**console answer** - Answer an incoming console call  
**console autoanswer** - Sets/displays autoanswer  
**console boost** - Sets/displays mic boost in dB  
**console dial** - Dial an extension on the console  
**console flash** - Flash a call on the console  
**console hangup** - Hangup a call on the console  
**console mute** - Disable mic input  
**console send text** - Send text to the remote device  
**console transfer** - Transfer a call to a different extension  
**console unmute** - Enable mic input

### Core related commands

**core clear profile** - Clear profiling info  
**core set debug channel** - Enable/disable debugging on a channel  
**core set debug** - Set level of debug chattiness  
**core set debug off** - Turns off debug chattiness  
**core set global** - Set global dialplan variable  
**core set verbose** - Set level of verbosity  
**core show applications** - Shows registered dialplan applications  
**core show application** - Describe a specific dialplan application  
**core show audio codecs** - Displays a list of audio codecs  
**core show channels** - Display information on channels  
**core show channel** - Display information on a specific channel  
**core show channeltypes** - List available channel types  
**core show channeltype** - Give more details on that channel type  
**core show codecs** - Displays a list of codecs  
**core show codec** - Shows a specific codec  
**core show config mappings** - Display config mappings (file names to config engines)  
**core show file formats** - Displays file formats  
**core show file version** - List versions of files used to build Asterisk  
**core show functions** - Shows registered dialplan functions  
**core show function** - Describe a specific dialplan function  
**core show globals** - Show global dialplan variables  
**core show hints** - Show dialplan hints  
**core show image codecs** - Displays a list of image codecs

**core show image formats** - Displays image formats  
**core show license** - Show the license(s) for this copy of Asterisk  
**core show profile** - Display profiling info  
**core show switches** - Show alternative switches  
**core show threads** - Show running threads  
**core show translation** - Display translation matrix  
**core show uptime** - Show uptime information  
**core show version** - Display version info  
**core show video codecs** - Displays a list of video codecs  
**core show warranty** - Show the warranty (if any) for this copy of Asterisk

### **Database commands**

**database del** - Removes database key/value  
**database deltree** - Removes database keytree/values  
**database get** - Gets database value  
**database put** - Adds/updates database value  
**database show** - Shows database contents  
**database showkey** - Shows database contents

### **Dialplan commands**

**dialplan add extension** - Add new extension into context  
**dialplan add ignorepat** - Add new ignore pattern  
**dialplan add include** - Include context in other context  
**dialplan reload** - Reload extensions and \*only\* extensions  
**dialplan remove extension** - Remove a specified extension  
**dialplan remove ignorepat** - Remove ignore pattern from context  
**dialplan remove include** - Remove a specified include from context  
**dialplan save** - Save dialplan  
**dialplan show** - Show dialplan

### **DUNDI commands**

**dundi debug** - Enable DUNDi debugging  
**dundi flush** - Flush DUNDi cache  
**dundi lookup** - Lookup a number in DUNDi  
**dundi no debug** - Disable DUNDi debugging  
**dundi no store history** - Disable DUNDi historic records  
**dundi precache** - Precache a number in DUNDi  
**dundi query** - Query a DUNDi EID  
**dundi show entityid** - Display Global Entity ID  
**dundi show mappings** - Show DUNDi mappings  
**dundi show peers** - Show defined DUNDi peers  
**dundi show peer** - Show info on a specific DUNDi peer



**dundi show precache** - Show DUNDi precache  
**dundi show requests** - Show DUNDi requests  
**dundi show trans** - Show active DUNDi transactions  
**dundi store history** - Enable DUNDi historic records

### **GTalk & Jabber commands**

**gtalk reload** - Enable Jabber debugging  
**gtalk show channels** - Show GoogleTalk Channels  
**jabber debug** - Enable Jabber debugging  
**jabber debug off** - Disable Jabber debug  
**jabber reload** - Enable Jabber debugging  
**jabber show connected** - Show state of clients and components  
**jabber test** - Shows roster, but is generally used for mog's debugging.

### **IAX2 commands**

**iax2 provision** - Provision an IAX device  
**iax2 prune realtime** - Prune a cached realtime lookup  
**iax2 reload** - Reload IAX configuration  
**iax2 set debug** - Enable IAX debugging  
**iax2 set debug jb** - Enable IAX jitterbuffer debugging  
**iax2 set debug jb off** - Disable IAX jitterbuffer debugging  
**iax2 set debug off** - Disable IAX debugging  
**iax2 set debug trunk** - Enable IAX trunk debugging  
**iax2 set debug trunk off** - Disable IAX trunk debugging  
**iax2 show cache** - Display IAX cached dialplan  
**iax2 show channels** - List active IAX channels  
**iax2 show firmware** - List available IAX firmwares  
**iax2 show netstats** - List active IAX channel netstats  
**iax2 show peers** - List defined IAX peers  
**iax2 show peer** - Show details on specific IAX peer  
**iax2 show provisioning** - Display iax provisioning  
**iax2 show registry** - Display IAX registration status  
**iax2 show stats** - Display IAX statistics  
**iax2 show threads** - Display IAX helper thread info  
**iax2 show users** - List defined IAX users  
**iax2 test losspect** - Set IAX2 incoming frame loss percentage

### **Manager commands**

**manager show command** - Show a manager interface command  
**manager show commands** - List manager interface commands  
**manager show connected** - List connected manager interface users  
**manager show eventq** - List manager interface queued events

**manager show users** - List configured manager users  
**manager show user** - Display information on a specific manager user

### **MGCP commands**

**mgcp audit endpoint** - Audit specified MGCP endpoint  
**mgcp reload** - Reload MGCP configuration  
**mgcp set debug** - Enable MGCP debugging  
**mgcp set debug off** - Disable MGCP debugging  
**mgcp show endpoints** - List defined MGCP endpoints

### **Module management**

**module load** - Load a module by name  
**module reload** - Reload configuration  
**module show** - List modules and info  
**module show like** - List modules and info  
**module unload** - Unload a module by name

### **PRI commands**

**pri debug span** - Enables PRI debugging on a span  
**pri intense debug span** - Enables REALLY INTENSE PRI debugging  
**pri no debug span** - Disables PRI debugging on a span  
**pri set debug file** - Sends PRI debug output to the specified file  
**pri show debug** - Displays current PRI debug settings  
**pri show spans** - Displays PRI Information  
**pri show span** - Displays PRI Information  
**pri unset debug file** - Ends PRI debug output to file

### **Queue commands**

**queue add member** - Add a channel to a specified queue  
**queue remove member** - Removes a channel from a specified queue  
**queue show** - Show status of a specified queue  
**rtcp debug ip** - Enable RTCP debugging on IP  
**rtcp debug** - Enable RTCP debugging  
**rtcp debug off** - Disable RTCP debugging  
**rtcp stats** - Enable RTCP stats  
**rtcp stats off** - Disable RTCP stats  
**rtp debug ip** - Enable RTP debugging on IP  
**rtp debug** - Enable RTP debugging  
**rtp debug off** - Disable RTP debugging

**say load** - Set/show the say mode  
**show parkedcalls** - Lists parked calls  
**show queue** - Show information for target queue  
**show queues** - Show the queues

### **SIP commands**

**sip history** - Enable SIP history  
**sip history off** - Disable SIP history  
**sip notify** - Send a notify packet to a SIP peer  
**sip prune realtime** - Prune cached Realtime object(s)  
**sip prune realtime peer** - Prune cached Realtime peer(s)  
**sip prune realtime user** - Prune cached Realtime user(s)  
**sip reload** - Reload SIP configuration  
**sip set debug** - Enable SIP debugging  
**sip set debug ip** - Enable SIP debugging on IP  
**sip set debug off** - Disable SIP debugging  
**sip set debug peer** - Enable SIP debugging on Peername  
**sip show channels** - List active SIP channels  
**sip show channel** - Show detailed SIP channel info  
**sip show domains** - List our local SIP domains.  
**sip show history** - Show SIP dialog history  
**sip show inuse** - List all inuse/limits  
**sip show objects** - List all SIP object allocations  
**sip show peers** - List defined SIP peers  
**sip show peer** - Show details on specific SIP peer  
**sip show registry** - List SIP registration status  
**sip show settings** - Show SIP global settings  
**sip show subscriptions** - List active SIP subscriptions  
**sip show users** - List defined SIP users  
**sip show user** - Show details on specific SIP user

### **Skinny commands**

**skinny reset** - Reset Skinny device(s)  
**skinny set debug** - Enable Skinny debugging  
**skinny set debug off** - Disable Skinny debugging  
**skinny show devices** - List defined Skinny devices  
**skinny show lines** - List defined Skinny lines per device

### **Voicemail commands**

**voicemail show users** - List defined voicemail boxes  
**voicemail show users for** - List defined voicemail boxes for target context  
**voicemail show zones** - List zone message formats

## Zaptel commands

**zap destroy channel** - Destroys a channel  
**zap restart** - Fully restart zaptel channels  
**zap show cadences** - List cadences  
**zap show channels** - Show active zapata channels  
**zap show channel** - Show information on a channel  
**zap show status** - Show all Zaptel cards status

6.

**Je travail actuellement au déploiement d'une solution Asterisk, seulement j'ai un petit problème entre mes deux serveurs Asterisk.**

**Asterisk1, peut appeler Asterisk 2, mais pas l'inverse. A noter que Asterisk 2 est géré par Vbox et que normalement tout est ok de son côté pour que ça marche... je pense donc que ça vient de chez moi.**

## **Voici mon Extensions.conf**

```
[general]
static=yes
writeprotect=no
language=fr

[mon_context]

include => trunk
include => appelEntrant

exten => 100,1,Dial(SIP/100,10)
exten => 100,2,Hangup()

exten => 101,1,Dial(SIP/101,10)
exten => 101,2,Hangup()

[trunk]

include => mon_context

exten => _3X.,1,SetCallerPres(allowed)
exten => _3X.,2,Set(CALLERID(num)=${CALLERID(num)})
exten => _3X.,3,Dial(SIP/${EXTEN:1}@asterisk2)
exten => _3X.,4,Hangup
exten => _3X.,104,Hangup
```

### **voici le sip.conf**

```
[general]
language=fr
disallow=all
allow=ulaw
allow=alaw

[asterisk2]
host=192.xxx.xxx.xxx
type=friend
fromuser=altitude_vbox_trunk
register=no
canreinvite=no
insecure=port,invite
qualify=yes
```

```
[100]
host=dynamic
type=friend
callerid="Poste A"
context=mon_context
nat=no
port=5060
insecure=port,invite
canreinvite=yes
call-limit=2
qualify=yes
dtmfmode=auto
```

### **7.**

télécharger le paquet Debian ~~ekiga~~ et les autres paquets dont il dépend sur le site <http://www.ekiga.org>. Les fichiers seront tous sauvegardés dans un répertoire ~~Ekiga~~ :

```
cd Ekiga
dpkg --install *.deb
```

Note : Pour la version de développement ~~ekiga-snapshot~~ contenue dans le paquet Debian

### **8.**

**En revanche les appels entrants me pose un probleme. Je recois bien le 'coup de file' de mon provider avec la presentation du numero mais je n'arrive pas a catcher l'extension qu'il faut ...**

**Plutot qu'un long discours voici les conf appels sortant et message debug.**

## sip.conf

```
[appelentrant]
type=peer
host=ip.provider
context=inbound
```

## extensions.conf

Est ce que l'erreur ne viendrait pas du fait de la reception d'un numero commençant par un +33 ??

```
exten => _+X.,1,Dial(SIP/moi)
exten => _+X.,2,Hangup
```

## 10.

I've been working on Load Testing some of our asterisk applications and thought I'd document the process here so I can find it when I need to do it again. There is a bit of setup to be done but when you get the whole picture is it's actually very easy. To do it you need at least 2 asterisk boxes - the one you are load testing (lets call it 'stressbox' for this example) and a another one to drive load to it which we'll call 'driver'.

We need to do the following steps:

- Create an sip account on stressbox for driver to connect to
- Register the account from the driver machine
- Create a dial plan which simulates a call
- Create a call file for the call
- Start you system monitoring
- Create lots of call files

So here is the step by step:

Create an sip account on stressbox for driver to connect to

In the sip.conf file on stressbox (the one we are testing) add the following:

```
[loadtest]
type=friend
username=loadtest
secret=1111
context=test
host=dynamic
```

Do a 'sip reload' in the asterisk console to register this change.

Register the account from the driver machine

Then to register the account from the driver machine add the following to sip.conf on driver:

```
register => loadtest:1111@xxx.xxx.xxx.xxx
```

Where xxx.xxx.xxx.xxx is the ip address of the stressbox server.

Your driver machine should now be able to make calls to the stressbox machine and you should see a line that looks like this if you were watching the console:

```
-- Registered SIP 'loadtest' at yyy.yyy.yyy.yyy port 9350 expires 120
```

Create a dial plan which simulates a call

Then on the driver machine we need to create a dial plan which simulates the call we want to do and place it in extensions.conf. Here's an example dial plan:

```

[dialplan-loadtest1]
exten => s,1,Answer
;wait for welcome to play
exten => s,n,Wait(10)
;enter username
exten => s,n,SendDTMF(1234)
exten => s,n,Wait(4)
;enter password
exten => s,n,SendDTMF(0000)
;listen to menu
exten => s,n,Wait(10)
;select menu option 1
exten => s,n,SendDTMF(1)
exten => s,n,Wait(3)
;select menu option 2
exten => s,n,SendDTMF(2)
exten => s,n,Wait(4)

;Play some audio
exten => s,n,Playback(sampleaudio1)
;Hangup
exten => s,n,Hangup

```

This example does a login to a phone system and then selects some options and plays some audio and hangs up. The dial plan can be as simple or as complex as you like.

Create a call file for the call

Last config step is to create a call file which will tell asterisk on driver to call asterisk on stressbox. Create a file test.call with the following:

```

Channel: SIP/6@xxx.xxx.xxx.xxx
MaxRetries: 1
RetryTime: 5
WaitTime: 10
Context: dialplan-loadtest1
Extension: s
Priority: 1

```

A bit of explanation of these lines: Channel: SIP/6@xxx.xxx.xxx.xxx - the 6 is the extension to dial on the stressbox server - change as necessary. The ip adress should be the stressbox one. Context: dialplan-loadtest1 - this is the dial plan that will be run for this call.

Start you system monitoring

If you are going to do some load testing then you probably want to monitor the load on the machine - it is recommended to record the load of all machines involved - i.e. stressbox and driver to ensure that driver is not the bottleneck in the load test.

For this I use a vmstat on linux via a little script I wrote which makes it easy to collect stats from multiple machines

It would be used as follows:

```
./multivmstat stressbox,driver 5
```

This will connect via ssh to each box and run vmstat on them every five seconds. This can be graphed later - I'll post the scripts for this shortly.

Create lots of call files

You are not ready to test you system. To do this you need to create copies the test.call file and move them into /var/spool/asterisk/outgoing on the driver machine.

I use a little script to do this:

```
#!/bin/sh
```

```
#usage: makecalls callfile count delaybetween
```

```
callfile=$1
```

```
count=$2
```

```
delaybetween=$3
```

```
callnumber=0
```

```
asteriskcalldir=/var/spool/asterisk/outgoing/
```

```
asteriskuser=asterisk
```

```
echo "Using $callfile for $count iterations with a $delaybetween second delay"
```

```
while [ $callnumber -lt $count ];
```

```
do
```

```
    callnumber=`expr $callnumber + 1`
```

```
    echo "Call Number: $callnumber"
```

```
    cp $callfile $callfile.tmp
```

```
    chown $asteriskuser:$asteriskuser $callfile.tmp
```

```
    mv $callfile.tmp $asteriskcalldir
```

```
    sleep $delaybetween
```

```
done
```

This is used as follows:

```
sudo ./makecalls test.call 5 2
```

This will make 5 calls with a 2 second delay between the start of each call.

Note: It needs to be run as sudo so it can change the ownership correctly.

Credits: Thanks to [Stuart Elvish](#) for his help with getting this working so nicely.

Happy load testing. Cheers, Mark