Asterisk Tutorial

Jonny Martin
Citylink

jonny@citylink.co.nz

Introduction

- An introduction to installing and configuring Asterisk
- Intermediate level - assumes basic knowledge of networking, linux systems, and VoIP
- We'll be building a real live Asterisk box as we progress through the slides
- If you have a question please ask
- Asterisk is the goods :)

Agenda 1/2

- Installing Asterisk
- All about Asterisk in three slides
- Telephony Hardware
- A basic Asterisk configuration
- Zaptel hardware configuration
- Asterisk codecs
- System dimensioning

Agenda 2/2

- Voicemail and conferencing
- Administering Asterisk
- Advanced Asterisk - DBs, AGI scripts, scaling
- Configuration files

What is Asterisk?

- Asterisk, *The* Open Source PBX. www.asterisk.org
- A complete PBX in software
- Runs on Linux, BSD, MacOSX, and others
- Covers most VoIP protocols
- Many features built in - voicemail, conferencing, IVR, queuing, as well as standard calling functions
- Highly extensible - can handle virtually any task imaginable
- Many different hardware telephony cards available

Asterisk History

- Originally developed by Mark Spencer starting around 1999
- He needed a flexible PBX for his linux support company so wrote one
- Realised once a call is inside a PC, *anything* can be done with it - hence the name Asterisk
- Met Jim Dixon from the Zapata telephony project in 2001 which provided hardware and a business model to further development
- Now an active Asterisk development community

Useful Reading

- Asterisk, The Future of Telephony. By Jared Smith, Jim Van Meggelen, Leif Madsen. ISBN: 0-596-00962-3
    - Published under Creative Commons license
    - http://www.asteriskdocs.org/modules/tinycontent/index.php?id=11
- www.voip-info.org
    - A public wiki - generally good information, but to be taken with a grain of salt
- www.asterisk.org
- www.digium.com

Installing Asterisk

- Asterisk uses three main packages:
  - asterisk
  - zaptel
  - libpri
- Compile Requirements:
  - GCC (version 3.x or later)
  - Kernel source
  - Kernel headers
  - bison
  - openssl, openssl-dev, libssl-dev
  - libnewt

Download Source

```
# cd /usr/src/
# wget --passive-ftp ftp.digium.com/pub/asterisk/asterisk-1.*.tar.gz
# wget --passive-ftp ftp.digium.com/pub/asterisk/asterisk-sounds-*.tar.gz
# wget --passive-ftp ftp.digium.com/pub/zaptel/zaptel-*.tar.gz
# wget --passive-ftp ftp.digium.com/pub/libpri/libpri-*.tar.gz


# tar zxvf zaptel-*.tar.gz
# tar zxvf libpri-*.tar.gz
# tar zxvf asterisk-*.tar.gz
# tar zxvf asterisk-sounds*.tar.gz
```

*** If using Linux kernel 2.4 a symbolic link named linux-2.4 is required pointing to your kernel source:**

```
#ln -s /usr/src/`uname -r` /usr/src/linux-2.4
```

Compile Zaptel

- Several features in Asterisk require an accurate timing source, e.g. conferencing
- Digium PCI hardware provides this 1kHz timing clock
- If you aren't using PCI hardware the *ztdummy* driver can be used
    - Kernels 2.4.5 and greater use the UHCI USB controller for this (so you need the *usb-uhci* module loaded)
    - The 2.6 kernel provides a 1kHz so a USB controller is not needed
- Need to uncomment out 'ztdummy' in Makefile

```
MODULES=zaptel tor2 torisa wcusb wcfxo wctdm \
        ztdynamic ztd-eth wct1xxp wct4xxp wcte11xp # ztdummy
```

Compile Zaptel

```
# cd /usr/src/zaptel-version
# make clean
# make
# make install
# make config
```

- Also installs some tools:
  - *ztcfg* - reads config in */etc/zaptel.conf* to configure hardware
  - *zttool* - for monitoring installed hardware
  - *ztmonitor* - for monitoring active channels
- zconfig.h contains many zaptel compile-time options - echo cancellation options, RAS options, etc.

Compile Libpri

```
# cd /usr/src/libpri-version
# make clean
# make
# make install
```

- Used by many manufacturers of PCI TDM cards
  - Safe to compile even if a card is not installed/used

Compile Asterisk

```
# cd /usr/src/asterisk-version
# make clean
# make
# make install
# make samples
```

The Easy Way

- Use pre-compiled binary packages
  - RPM packages for redhat
  - DEB packages for Debian
  - Asterisk.pkg for MacOSX http://www.astmasters.net
- I'll be using debian .deb packages for this tutorial
  - Latest debian package is Asterisk v 1.0.7
  - CVS head 1.2.4

The Easier Way

- Pop an asterisk@home live CD in a machine and go for it!
- http://asteriskathome.sourceforge.net/
- Too easy for this tutorial :)
- Very sophisticated system
    - A lot of integration work to provide billing and GUI management
    - Well worth trying

Debian install

```
apt-get install asterisk
apt-get install zaptel
apt-get build-dep asterisk
apt-get install kernel-headers-`uname -r`
ln -s /usr/src/kernel-headers-`uname -r`/ /usr/src/linux
m-a build zaptel
dpkg -i zaptel-modules-xxxxxx.deb
depmod
modprobe zaptel
modprobe wcte11xp   # if using TE110P single span T1/E1 card
modprobe wcfxo      # if using single port FXO card
modprobe ztdummy    # if using ztdummy
ztcfg
zttool
```

*** To get ztdummy, modify Makefile to uncomment 'ztdummy'**
*** On Debian, add 'ztdummy' to /etc/module to get ztdummy to load at boot**

Compile mpg123

- Required to stream music on hold
- Must use version mpg123 version 0.59r as others don't work
- http://www.mpg123.de/cgi-bin/sitexplorer.cgi?/mpg123/

```
# cd /usr/src
# wget http://www.mpg123.de/mpg123/mpg123-0.59r.tar.gz
# tar -zxvf mpg123-0.59r.tar.gz
# cd mpg123-0.59r
# make clean
# make linux-devel
# make install
# ln -s /usr/local/bin/mpg123 /usr/bin/mpg123   # this is where asterisk looks
```

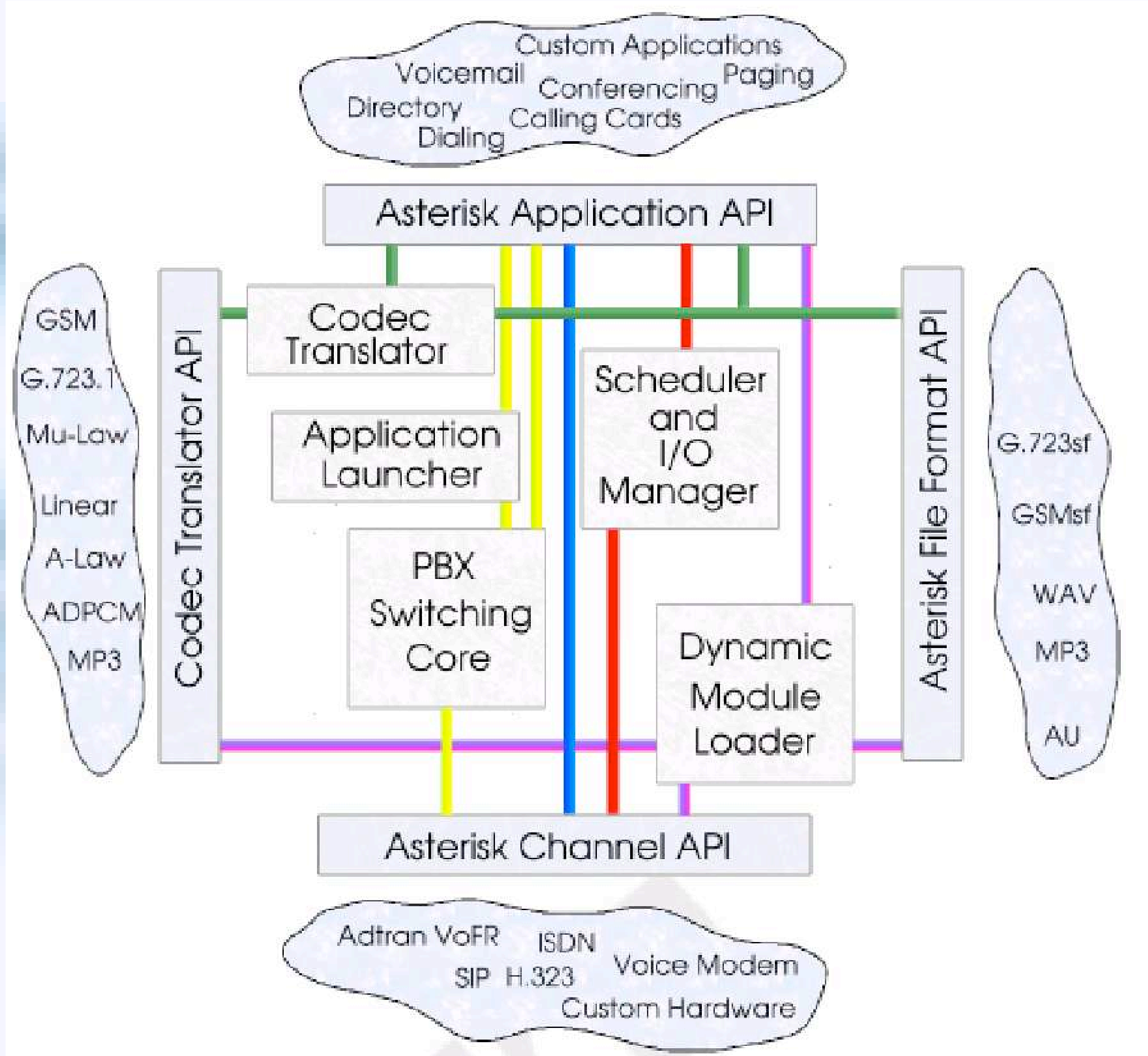
```
jonny@collins:~# asterisk -h
Asterisk 1.0.7-BRIstuffed-0.2.0-RC7k, Copyright (C) 2000-2004, Digium.
Usage: asterisk [OPTIONS]
Valid Options:
   -V              Display version number and exit
   -C <configfile> Use an alternate configuration file
   -G <group>      Run as a group other than the caller
   -U <user>       Run as a user other than the caller
   -c              Provide console CLI
   -d              Enable extra debugging
   -f              Do not fork
   -g              Dump core in case of a crash
   -h              This help screen
   -i              Initialize crypto keys at startup
   -n              Disable console colorization
   -p              Run as pseudo-realtime thread
   -q              Quiet mode (suppress output)
   -r              Connect to Asterisk on this machine
   -R              Connect to Asterisk, and attempt to reconnect if disconnected
   -t              Record soundfiles in /var/tmp and move them where they belong after
they are done.
   -v              Increase verbosity (multiple v's = more verbose)
   -x <cmd>        Execute command <cmd> (only valid with -r)

jonny@collins:~# asterisk -r
Asterisk 1.0.7-BRIstuffed-0.2.0-RC7k, Copyright (C) 1999-2004 Digium.
Written by Mark Spencer <markster@digium.com>
=========================================================================
Connected to Asterisk 1.0.7-BRIstuffed-0.2.0-RC7k currently running on collins (pid =
10763)
collins*CLI>
```

Asterisk File Locations

- */etc/asterisk/* - Asterisk configuration files
- */usr/lib/asterisk/modules/* - all loadable modules: codecs, channels, formats etc.
- */var/lib/asterisk/* - contains the *astdb*, sounds, images, firmware and keys
- */var/spool/asterisk/* - temporary files and voicemail files
- */var/run/* - contains the process ID (PID) for running processes, including Asterisk
- */var/log/asterisk/* - Asterisk log files
- */var/log/asterisk/cdr-csv/* - Asterisk call detail records

Asterisk Basics

- Asterisk is a hybrid TDM and packet voice PBX
- Interfaces any piece of telephony hardware or software to any telephony application
- Prime components: channels and */etc/asterisk/extensions.conf* - the Asterisk dial plan
- Channels can be many different technologies - SIP, IAX, H323, skinny, Zaptel, and others as they are created
- *extensions.conf* is basically a powerful programming language controlling the flow of calls
- Applications do the work - answering a channel, ringing a channel, providing a voicemail system etc.


Custom Applications
Voicemail
Paging
Conferencing
Directory
Calling Cards
Dialing
Asterisk Application API
GSM
G.723.1
Mu-Law
Linear
A-Law
ADPCM
MP3
Codec Translator API
Codec Translator
Application Launcher
PBX Switching Core
Scheduler and I/O Manager
Dynamic Module Loader
Asterisk File Format API
G.723sf
GSMsf
WAV
MP3
AU
Asterisk Channel API
Adtran VoFR
ISDN
SIP
H.323
Voice Modem
Custom Hardware

Telephony Hardware

- Digium make several digital and analog PCI cards
  - T1 / E1 single to quad span cards
  - FXO and FXS interfaces up to 24 ports
  - One port FXO card - PCI Intel Winmodem
- www.digium.com
- Plus the usual array of SIP and IAX phones and analogue adapters (ATAs)
- Even interfaces to proprietary digital key phones are available

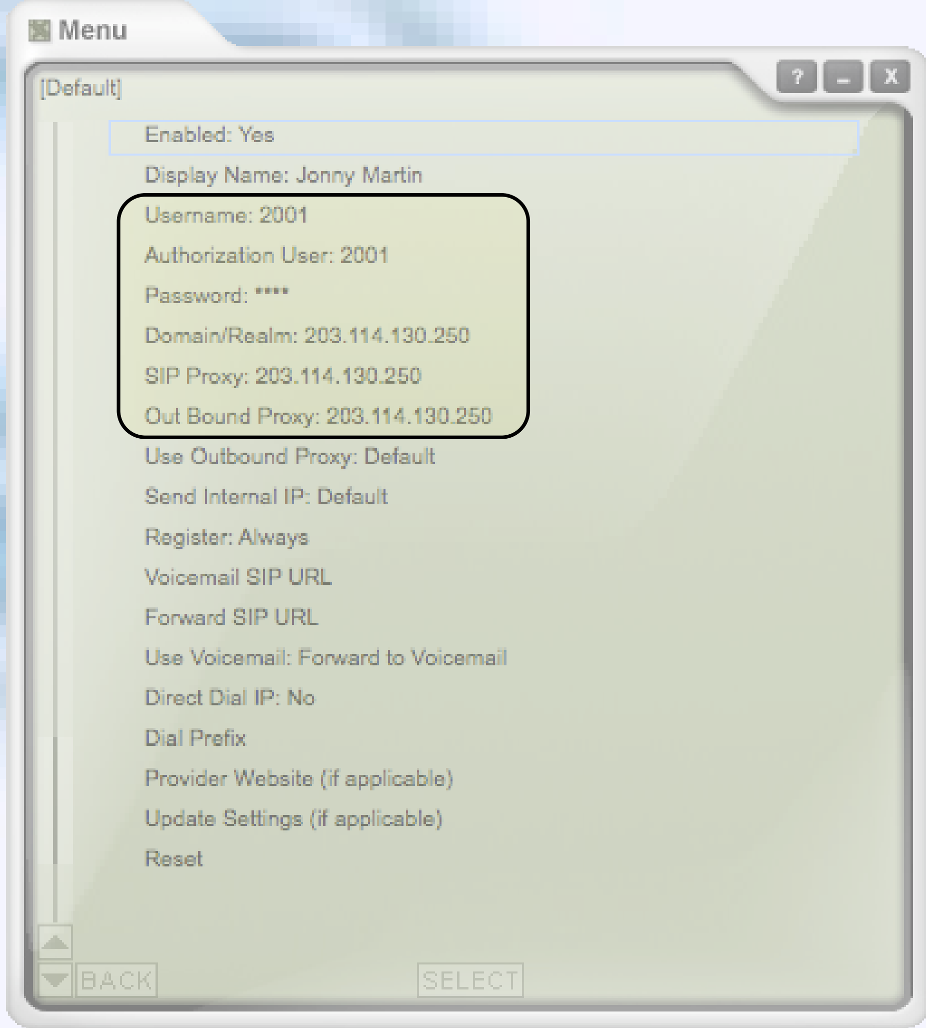
Basic System Configuration

- Two SIP devices: a WiFi phone and a softphone on a laptop
- SIP gateway for calls to the PSTN
- Will be working with *sip.conf* and *extensions.conf*
- Simple dial plan:
  - softphone (SIP user 2001, pw j0nny), extension 2001
  - wifi phone (SIP user 2002, pw whyfry), extension 2002
  - echo test, extension 500
  - send all other calls to gateway
  - inbound calls from the gateway to (+64 4) 4980007 to ring extension 2001

Setup SIP endpoints

- Using the Xten X-lite softphone
  - Download at http://www.xten.com/index.php?menu=download
- Need to set SIP username and password, and SIP server
  - Main Menu > System Settings > SIP Proxy > Default

Setup SIP endpoints



```
/etc/asterisk/sip.conf
[general]
context=default                 ; Default context for incoming calls
port=5060                       ; UDP Port to bind to (SIP standard port is 5060)
bindaddr=0.0.0.0                ; IP address to bind to (0.0.0.0 binds to all)
srvlookup=yes                   ; Enable DNS SRV lookups on outbound calls


[2001]
type=friend                     ; both send and receive calls from this peer
host=dynamic                    ; this peer will register with us
username=2001
secret=j0nny
canreinvite=no                  ; don't send SIP re-invites (ie. terminate rtp stream)
nat=yes                         ; always assume peer is behind a NAT
context=phones                  ; send calls to 'phones' context
dtmfmode=rfc2833                ; set dtmf relay mode
allow=all                       ; allow all codecs

[2002]
type=friend
host=dynamic
username=2002
secret=whyfry
canreinvite=no
nat=yes
context=phones
dtmfmode=rfc2833
allow=all
```

/etc/asterisk/sip.conf ctd...

```
[wlg-gateway]
type=friend
disallow=all
allow=ulaw
context=from-wlg-gateway
host=202.7.4.40
canreinvite=no
dtmfmode=rfc2833
allow=all
```

/etc/asterisk/extensions.conf

```
[general]
static=yes              ; default values for changes to this file
writeprotect=no         ; by the Asterisk CLI

[globals]
; variables go here

[default]
; default context

[phones]
; context for our phones
exten => 2001,1,Dial(SIP/2001)
exten => 2002,1,Dial(SIP/2002)

exten => 500,1,Answer()
exten => 500,2,Playback(demo-echotest)       ; Let them know what's going on
exten => 500,3,Echo                          ; Do the echo test
exten => 500,4,Playback(demo-echodone)       ; Let them know it's over
exten => 500,5,Hangup

exten => _.,1,Dial(SIP/${EXTEN}@wlg-gateway)  ; match anything and send to wlg-gateway
exten => _.,2,Hangup

[from-wlg-gateway]
; context for calls coming from wlg-gateway

exten => 4980007,1,Dial(SIP/2001&SIP/2002)
exten => _.,1,Congestion()                     ; everyone else gets congestion
```

Dial Plan Basics - Contexts

- *extensions.conf* split into sections called *contexts*

  [context-name]

- *contexts* isolated from one another - can have the same extension in multiple contexts
- Calls from a channel land in the context specified by that channel,
  - Calls land in default context if nothing is specified
  - Be careful with what is in the default context - it is easy to give access to more than is intended

Dial Plan Basics - Extensions

- One or more extensions in each context
- An extension is followed by an incoming call or digits dialled on a channel

  exten => name,priority,application()

  exten => 2001,1,Dial(SIP/2001)

- Priorities are numbered and followed sequentially from '1'
  - Asterisk will stop processing an extension if you skip a priority
- Each priority executes one specific application

Dial Plan Basics - Applications

- Applications are what ‘do things’ in the Asterisk dial plan
  - play a sound
  - answer a call
  - collect dtmf digits
  - interact with a database
- Can take zero or more arguments
  - Answer()
  - Dial(SIP/2001)
  - AnApplicationWithThreeArguments(arg1,arg2,arg3)
- Arguments can be seperated with a pipe (|) or a comma.

Dial Plan Basics - Variables

- Three types of variables available in the dial plan.
- Global
  - Set in the *[globals]* section of *extensions.conf*
- Channel
  - Variables set using the *set* command on a per channel basis
  - A number of pre-defined channel variables - e.g. ${EXTEN}
- Environment
  - Access to UNIX environment variables from within Asterisk

Dial Plan Basics - Variables

- Some of the pre-defined channel variables:

  ${CALLERID}
  ${CALLERIDNAME}
  ${CALLERIDNUM}
  ${CHANNEL}
  ${CONTEXT}
  ${EXTEN}
  ${SIPUSERAGENT}

Let's Add To Our System

- Introduce a global variable: ${jonnysphone}
- Ring phones for 15sec and divert to voicemail if unanswered
- If our phones are busy, divert to voicemail
- Only allow Wellington NZ numbers (04xxxxxxx) to be dialled out gateway
- Add a 'hangup' extension ('h' extension) to ensure asterisks hangs up calls when finished

/etc/asterisk/extensions.conf

```
[general]
static=yes              ; default values for changes to this file
writeprotect=no         ; by the Asterisk CLI

[globals]
JONNYSPHONE=SIP/2001

[default]
; default context

[phones]
; context for our phones
include => fun-stuff    ; include another context's extensions here
include => gateway      ;

exten => 2001,1,Dial(${JONNYSPHONE},15)
exten => 2001,2,Voicemail(u${JONNYSPHONE}@${CONTEXT})
exten => 2001,102,Voicemail(b{JONNYSPHONE}@${CONTEXT})

exten => 2002,1,Dial(SIP/2002,15)
exten => 2002,2,Voicemail(u2002@phones)
exten => 2002,102,Voicemail(b2002@phones)

exten => h,1,Hangup
```

/etc/asterisk/extensions.conf ctd...

```
[fun-stuff]
exten => 500,1,Answer()
exten => 500,2,Playback(demo-echotest)  ; Let them know what's going on
exten => 500,3,Echo                     ; Do the echo test
exten => 500,4,Playback(demo-echodone)  ; Let them know it's over
exten => 500,5,Hangup

[gateway]
exten => _04NXXXXXX,1,Dial(SIP/${EXTEN}@wlg-gateway)
exten => _04NXXXXXX,2,Hangup

exten => _104NXXXXXX,1,Dial(SIP/${EXTEN:1}@wlg-gateway)  ; strip one and send out
exten => _104NXXXXXX,2,Hangup

[from-wlg-gateway]
; context for calls coming from wlg-gateway
exten => 4980007,1,Dial(SIP/2001&SIP/2002)
exten => _.,1,Congestion()              ; everyone else gets congestion
```

Dial Plan Pattern Matching

exten => _04NXXXXXX,1,SomeApplication()

exten => _.,1,SomeApplication()

- _ denotes a pattern matching extension
- N matches any number from 2 through 9
- X matches any single digit
- . matches one or more of any digit
- [2-6] matches any of 2,3,4,5,6

Zaptel Interfaces

- Two configuration files:
  - */etc/zaptel.conf* - low level configuration for the hardware interface
  - */etc/asterisk/zapata.conf* - configuration for Asterisk's interface to the hardware
- In *zaptel.conf* the comment character is the hash (#)
- In all other config files the comment character is the semi-colon (;) as a hash is a valid telephone digit

/etc/zaptel.conf

```
# Zaptel Configuration File
#
# This file is parsed by the Zaptel Configurator, ztcfg
#
#
# First come the span definitions, in the format
# span=<span num>,<timing>,<line build out (LBO)>,<framing>,<coding>[,yellow]
#
# The framing is one of "d4" or "esf" for T1 or "cas" or "ccs" for E1
# The coding is one of "ami" or "b8zs" for T1 or "ami" or "hdb3" for E1
# E1's may have the additional keyword "crc4" to enable CRC4 checking
#
# Next come the definitions for using the channels.  The format is:
# <device>=<channel list>
#
# 10 channel E1
span=1,0,0,ccs,hdb3,crc4
bchan=1-10
dchan=16

# if we had some FXO interfaces we would uncomment this
#fxsks=32
#fxsks=33

# Load tones for specific country
loadzone = nz
#loadzone = us-old
defaultzone=nz
```

/etc/asterisk/zapata.conf

```
[trunkgroups]
; Trunk groups are used for NFAS or GR-303 connections.
; Spanmap: Associates a span with a trunk group
;        spanmap => <zapspan>,<trunkgroup>[,<logicalspan>]

[channels]
; Default language
;language=en
; Default context
context=default

; Signalling method (default is fxs).  Some of the more common values:
; em:      E & M
; em_w:    E & M Wink
; fxs_ks:  FXS (Kewl Start)
; fxo_ks:  FXO (Kewl Start)
; pri_cpe: PRI signalling, CPE side
; pri_net: PRI signalling, Network side
;
; Enable echo cancellation
; Use either "yes", "no", or a power of two from 32 to 256
echocancel=yes
echocancelwhenbridged=yes
```

/etc/asterisk/zapata.conf ctd...

```
; FXO example
;
signalling=fxs_ks ; X100P
echocancel=yes
echocancelwhenbridged=yes
echotraining=400
group=2
context=fxo1-incoming
channel => 32

; E1 PRI example

signalling=pri_cpe
switchtype=euroisdn
echocancel=128
echocancelwhenbridged=yes
echotraining=200
callerid=asreceived
;rxgain=-4 ; if needed
;txgain=-4 ; if needed
group=1
context=from-pri
channel => 1-10
```

Zaptel Channels

- Can dial a group of channels

```
exten => _.,1,Dial(g1/${EXTEN})
```

- Or dial a specific channel

```
exten => _.,1,Dial(4/${EXTEN})
```

The Start ‘s’ Extension

- The standard extension a call starts in without needed to specifically match an extension
- Often used with FXS/FXO cards due to lack of end to end signalling with analogue channels

```
[incoming]
exten => s,1,Answer()
exten => s,2,Background(enter-ext-of-person)
exten => 1,1,Playback(digits/1)
exten => 1,2,Goto(incoming,s,1)
exten => 2,1,Playback(digits/2)
exten => 2,2,goto(incoming,s,1)
exten => 3,1,Hangup
```

Other Standard Extensions

- i : Invalid
- s : Start
- h : Hangup
- t : Timeout
- T : AbsoluteTimeout
- o : Operator

Asterisk Codecs

Codec	Data bitrate (kbps)	Licence required?
G.711	64 kbps	No
G.726	16, 24, or 32 kbps	No
G.723.1	5.3 or 6.3 kbps	Yes (no for passthrough)
G.729A	8 kbps	Yes (no for passthrough)
GSM	13 kbps	No
iLBC	13.3 kbps (30-ms frames) or 15.2 kbps (20-ms frames)	No
Speex	Variable (between 2.15 and 22.4 kbps)	No

Server Dimensioning

- Many factors come into play, but in general the faster and the more RAM the better
- Running compressed codecs and echo cancellation takes up a lot of processor power
- Intel processors seem to perform better than AMD

Purpose	Number of channels	Minimum recommended
Hobby system	No more than 5	400-MHz x86, 256 MB RAM
SOHO[a] system	5 to 10	1-GHz x86, 512 MB RAM
Small business system	Up to 15	3-GHz x86, 1 GB RAM
Medium to large system	More than 15	Dual CPUs, possibly also multiple servers in a distributed architecture

Working With NAT

- NAT causes issues with SIP packets as endpoint IP addressing is embedded in packets
- Not using SIP re-invites helps a lot but at the expense of terminating the RTP media stream on the Asterisk box
- In *sip.conf* the line nat=yes tells Asterisk always to assume the peer may be behind a NAT

Voicemail

- Comedian Mail - a fully functional voicemail system included with Asterisk
- Supports busy and unavailable messages
  - exten => 2001,1,Voicemail(b2001)
  - exten => 2001,1,Voicemail(u2001)
- Voicemail can be emailed out a .wav attachment to users
- Standard IVR voicemail access
  - exten => 510,1,VoicemailMain

Voicemail

```
[general]

[default]
1234 => 4242,Example Mailbox,root@localhost

[phones]
2001 => 9999,Jonny Laptop,jonny@citylink.co.nz
2002 => 9999,Wifi Phone,jonny@citylink.co.nz
```

MeetMe Conferencing

```
/etc/asterisk/meetme.conf
; Configuration file for MeetMe simple conference rooms
; for Asterisk of course.
;
[rooms]
; Usage is conf => confno[,pin]
;
conf => 101,1234
conf => 102,2345

/etc/asterisk/extensions.conf
exten => 5101,1,Meetme(101|M)
exten => 5102,2,Meetme(102|M)
```

Music On Hold

- mpg123 player used to stream mp3s to a channel
- Can also stream a ShoutCast stream
- Use the line-in on a sound card in the Asterisk box for live audio
- mp3s must be converted to 8kHZ mono

```
exten => 501,1,WaitMusicOnHold(30)
```

Plays music on hold for 30 seconds.

Music on Hold

```
/etc/asterisk/musiconhold.conf
[classes]
; on Debian boxes files are in /usr/share/asterisk/mohmp3
; on other boxes, files are in /var/lib/asterisk/mohmp3
default => quietmp3:/usr/share/asterisk/mohmp3
loud => mp3:/usr/share/asterisk/mohmp3
podcasts => mp3:/usr/share/asterisk/mohmp3/podcasts
```

Console Commands

- Similar to IOS:
  - sip show peers
  - reload
  - ? for help, tab for command autocomplete
- Restart commands
  - restart gracefully: Restart Asterisk gracefully
  - restart now: Restart Asterisk immediately
  - restart when convenient: Restart Asterisk at empty call volume
  - reload: Reload configuration
  - stop gracefully: Gracefully shut down Asterisk
  - stop now: Shut down Asterisk imediately
  - stop when convenient: Shut down Asterisk at empty call volume

Console Commands

- sip debug: Enable SIP debugging
- sip no debug: Disable SIP debugging
- sip reload: Reload sip.conf (added after 0.7.1 on 2004-01-23)
- sip show channels: Show active SIP channels
- sip show channel: Show detailed SIP channel info
- sip show inuse: List all inuse/limit
- sip show peers: Show defined SIP peers (clients that register to your Asterisk server)
- sip show registry: Show SIP registration status (when Asterisk registers as a client to a SIP Proxy)
- sip show users: Show defined SIP users

Asterisk Database

- astdb - simple database forms part of Asterisk
- Dial plan and CLI can insert and remove data
- Data stored in a file, so is retained across Asterisk reloads and server reboots
- Data stored in groupings of *families* containing *keys*
- Applications
  - DBdel: Delete a key from the database
  - DBdeltree: Delete a family or keytree from the database
  - DBget: Retrieve a value from the database
  - DBput: Store a value in the database

Asterisk Database

```
; start counting and store count progress in astdb

exten => 510,1,Set(COUNT=${DB(test/count)})
exten => 510,2,SayNumber(${COUNT})
exten => 510,3,SetVar(COUNT=$[${COUNT} + 1]
exten => 510,4,Set(DB(test/count)=${COUNT})
exten => 510,5,Goto(1)
exten => 510,102,Set(DB(test/count)=1)
exten => 510,103,Goto(1)
```

Asterisk AGI Scripts

- Asterisk Gateway Interface
- Dial plan can call Perl, Python, PHP scripts
  - AGI script reads from STDIN to get information from Asterisk
  - AGI script writes data to STDOUT to send information to Asterisk
  - AGI script can write to STDERR to send debug information to the console
- Scripts stored in /usr/share/asterisk/agi-bin/ on Debian
- exten => 520,1,AGI(agi-script.agi)

answer: Asserts answer
channel status: Returns status of the connected channel
control stream file: Send the given file, allowing playback to be controled by the given digits, if any. (Asterisk 1.2)
database del: Removes database key/value
database deltree: Removes database keytree/value
database get: Gets database value
database put: Adds/updates database value
exec: Executes a given Application. (Applications are the functions you use to create a dial plan in extensions.conf).
get data: Gets data on a channel
get option: Behaves similar to STREAM FILE but used with a timeout option. (Asterisk 1.2)
get variable: Gets a channel variable
hangup: Hangup the current channel
noop: Does nothing
receive char: Receives one character from channels supporting it
receive text: Receives text from channels supporting it
record file: Records to a given file

say alpha: Says a given character string (Asterisk 1.2)
say date: Say a date (Asterisk 1.2)
say digits: Says a given digit string
say number: Says a given number
say phonetic: Say the given character string.
say time: Say a time
send image: Sends images to channels supporting it
send text: Sends text to channels supporting it
set autohangup: Autohangup channel in some time
set callerid: Sets callerid for the current channel
set context: Sets channel context
set extension: Changes channel extension
set music: Enable/Disable Music on hold generator, example "SET MUSIC ON default"
set priority: Prioritizes the channel
set variable: Sets a channel variable
stream file: Sends audio file on channel
verbose: Logs a message to the asterisk verbose log
wait for digit: Waits for a digit to be pressed

Scaling

- Scaling Asterisk normally involves multiple boxes
- Split off functionality
  - Conference server
  - SIP registration server
  - Use a central SIP proxy allow individual Asterisk boxes to query each other

Questions?