The Hitchhiker's Guide to Asterisk

Leif Madsen

Jared Smith

Steven Sokol

The Hitchhiker's Guide to Asterisk
by Leif Madsen, Jared Smith, and Steven Sokol

A guide to the basics of using Asterisk

Revision History

Revision 0.1 $Date: 2003/12/12 19:37:01 $

Table of Contents

Introductory letter from Mark Spencer (hopefully?)

Introductory letter from Mark Spencer (hopefully?)

Chapter 1. Introduction to Asterisk

General Concept of Asterisk

Asterisk: The Swiss Army Knife of Telephony

PBX, IVR, ACD

Telephony 101

Basic Conecpts (FXO/FXS, loop/ground start/PRI, etc.)

Telephony Resources: Newton's Telecom Dictionary, etc.

What to expect

Asterisk is not a turnkey system

Don't like it? Change it yourself!

Free and Open Source Software: GPL and LGPL Licensing

Asterisk Architecture

The Big Picture

Channels

Codecs and Conversions

Etc.

Key Components

Asterisk Software

Asterisk (Main PBX & Channels)

Zaptel (Drivers for Zaptel Hardware)

Libpri (ISDN PRI Drivers for Zaptel)

Zaptel Hardware

Overview

X100P - Single Port FXO Line Interface

S100U - Single Port FXS USB Interface

TDM400P - 4 Port FXS Analog Interface

T100P - Single Span T1/E1 Interface

T400P, E400P, TE410P, and TE405P - Quad-Span T1/E1 Interface

Channels

Zaptel Devices/Channels

The IAX/IAX2 Protocol

SIP

MGCP

H323

Skinny

Applications

Dial and Other Basics

Voicemail

Dial-Plan Scripting

Call Detail Recording (CDR)

Extensibility

AGI

Custom Applications

Add-On/Optional Components

Software

Soft Phones

Gnophone

iaxClient/iaxComm

DIAX

X-Lite/Pro

Management Tools

Astman/Gastman

Ethereal Plugin for IAX2

GUI/Web configuration tools

Gastman

Open H.323

Hardware

VoIP Hard Phones

VoIP Gateways

Channel Banks

Legacy PBX Systems

Other hardware options

VoiceTronix OpenLine and OpenSwitch Cards

QuickNet Cards

ISDN/CAPI Cards (Eicon, etc.)

Integrating ISDN channels to * can be done by several ways. Basically isdn4linux support is implemented in Asterisk. So called chan_modem_i4l. Another way is trough the powerfull CAPI interface. chan_capi is developed under the terms of the GPL an maintained by "Sir Kapejod". It is highly recommended to use chan_capi if your card is supported, because chan_capi supports even more functions than chan_modem_i4l. Because of this, this documentation is currently only written for chan_capi. All cards are welcome which come with native LINUX CAPI drivers.

With its features you are able to build your own ISDN-Box. This is a list of those implemented in chan_capi:

- ISDN connection handling (CID,DNID)
- multiple Controller support
- digital audio support
- DTMF detection/generation
- incoming/outgoing calls
- CLIP/CLIR
- early B3 connects
- native ISDN indications
- CD, HOLD, RETRIEVE, ECT
- overlap sending (dialtone)
- DID on P2P
- call progress (INFO_IND)
- RX/TX gains
- call deflection on circuit busy

Dialogic Cards (and Proprietary Drivers)

VoIP Service Providers

IAX providers

SIP providers

Chapter 2. Installing Asterisk

Requirements

PC Hardware Requirements

SOHO/Residential System

blah

Small Business System

blah

Medium Business/Small Call-center System

blah

Enterprise System

blah

VoIP Carrier System

blah

Linux Requirements

Tested Linux Distributions

Minimal Kernel Version

Required Packages

Hardware Installation

IRQ Sharing Issues

Digium Cards

ISDN Cards

ISDN Hardware must not be expensive. A basic AVM card that comes with CAPI compatible kernel modules is available for about 40$. But there are several differents between the capacity of the cards (think of more than 2 B-channels) and of different ISDN standards. chan_capi is programmed to work even with multiple ISDN cards. To use chan_capi you must have CAPI support in your kernel config and for your ISDN card. Mostly the vendor of your card is serving the newest drivers for their cards. So Eicon and AVM. It is always a bit tricky to get CAPI support working for different ISDN cards. Below are some links to good descriptions how to get CAPI support for several cards.

For a "Eicon Diva" visit:

```
http://www.melware.de/de/index.html
http://isdn4linux.org/~armin/divas/
http://www.eicon.com/worldwide/products/WAN/cn4linux.htm
```

For AVM cards visit:

http://www.avm.de

For AVM-Fritz! card:

ftp://ftp.avm.de/cardware/fritzcrd.pci/linux/suse.82/ (not only for SuSE) http://www.linux-magazin.de/Artikel/ausgabe/2000/10/Capi/capi.html

Other Cards (LineJack/PhoneJack/VoiceTronix/Dialogic)

Downloading Asterisk from CVS

What is CVS?

[CVS allows you to "check out" the latest version of a developer's code.]

The Asterisk "Versioning" Issues

[Asterisk CVS is often unstable, but until there are regular stable releases, it's about the best you can hope for.]

Your Initial Download

The most common way that people get Asterisk is through CVS. There are builds every once in a while, but they become outdated quite quickly. Your best bet is to get Asterisk from the Digium CVS server and compile it on your box. Before you start compiling, make sure that your system has these packages:

```
- readline and the readline development packages (on RedHat, it's
readline and readline-devel)
```

- openssl and openssl development packages (on RedHat, it's openssl and openssl-devel)
- Linux kernel 2.4.x and the Linux Kernel Source package (on RedHat, it's kernel-source)

If you are using a RedHat based system, when you install, just select the Development packages and it should pretty much give you everything you need to compile software with. You are also going to need CVS installed, but again, it is part of the development packages.

Now we can get the files from the CVS server.

```
cd /usr/src
export CVSROOT=:pserver:anoncvs@cvs.digium.com:/usr/cvsroot
cvs login  - the password is anoncvs
cvs checkout zaptel libpri asterisk
```

Your server will download all the appropriate files from the CVS server and place them in their respective directories (the top ones being zaptel, libpri and asterisk).

Updates

Compiling Asterisk

Using "make"

Now we need to compile Asterisk as root :

```
cd zaptel
make clean ; make install
cd ../libpri
make clean ; make install
cd ../asterisk
make clean ; make install
```

Depending on how fast your machine is, this will take a few minutes. On my Celeron 633 with 256 MB of RAM, this takes about 10-15 minutes. At the end of the configuration it will ask you if you want to make samples, don't bother with that, as we are going to make all the files we need from scratch (it's easier then trying to move stuff around in the sample files to fit our system)

If you want to install the asterisk initscript in **/etc/rc.d/init.d/** then you can type **make clean** in your **/usr/src/asterisk/** directory. Then you can start and stop Asterisk quite easily by doing **/etc/rc.d/init.d/asterisk {start|stop|reload|restart|status}**

Compiling the software

Zaptel

Libpri

Asterisk

Making the Samples/Demo

Making Code Documentation (Doxygen)

Why build code documentation?

What is Doxygen?

Code Documentation Layout

Building additional modules

H323 - McNamara

H323 - Manousos

MySQL CDR

CAPI/ISDN

The complete source code is available from kapejods website

http://www.junghanns.net/asterisk/downloads/chan_capi.0.3.0.tar.gz[7]

copy the sources to a folder of your choice and type the following commands to untar the source and change into its directory tree.

```
tar zxf chan_capi.0.3.0.tar.gz
cd chan_capi-0.3.0/
```

Now edit the file Makefile with your favorite editor to set it to your needs. First set the path to your asterisk include files.

"ASTERISK_HEADER_DIR=/usr/include/asterisk # standard path"

Then you can set some buildtime configuration parameters like early B3 connects, DEFLECT_ON_CIRCUITBUSY or software dtmf detection/generation. If everything is done simply save the file.

to compile and install the driver type:
make && make install

to install a sample capi.conf in asterisks conf dir:
make config

After this setup add in /etc/asterisk/modules.conf

load => chan_capi.so*

and in the [global] section:

"chan_capi.so=yes"

After these steps your channel-module is available in * but it has to be configured. This is done in the main CAPI configfile capi.conf.

Common Build Errors / Warnings

Via C3 is *NOT* an i686 processor

Building on a little-endian system

Loading Drivers

Linux Kernel Loadable Modules

Using "modprobe"

Adding zaptel modules to your startup file

Starting Asterisk

Manually starting Asterisk and the CLI

To start asterisk in the background; **/usr/sbin/asterisk** To start asterisk in console mode: **/usr/sbin/asterisk -c** To start asterisk in console mode with 3 levels of verbose: **/usr/sbin/asterisk -cvvv** To start asterisk in console mode, verbose and in debug mode: **/usr/sbin/asterisk -cvvvd** To start asterisk in console mode, verbose, debug and dump cores: **/usr/sbin/asterisk -cvvvgd**

Starting Asterisk using safe_asterisk

You can start Asterisk as a daemon using the safe_asterisk script located in /usr/sbin/

/usr/sbin/safe_asterisk

Accssing the CLI when Asterisk is running

If your asterisk is already running, you can reattach with the **-r** switch.

Logging/Tracing and Verbosity

Configuring Autostart w/ safe_asterisk

Linux Runlevels (the init sequence)

Modifying the startup manually

Notes

1. http://www.melware.de/de/index.html
2. http://isdn4linux.org/~armin/divas/
3. http://www.eicon.com/worldwide/products/WAN/cn4linux.htm
4. http://www.avm.de
5. ftp://ftp.avm.de/cardware/fritzcrd.pci/linux/suse.82/
6. http://www.linux-magazin.de/Artikel/ausgabe/2000/10/Capi/capi.html
7. http://www.junghanns.net/asterisk/downloads/chan_capi.0.3.0.tar.gz

Chapter 3. Basic Asterisk Configuration

[Some Basic Configuration Advice Here]

File Layout (The Asterisk Directory Structure)

Editing .conf files

Configuring Phones & Channels

The zapata.conf File

[All About zaptel/tormenta configuration]

The sip.conf File

The General Section

Supported Codecs

Registration of "peers"

SIP Device Entries

Unsupported codecs/G729.a/b

The oh323.conf File

Anybody Know This Well?

The skinny.conf File

Anybody Know This Well?

CAPI/ISDN

In the file capi.conf in your current asterisk config directory, you should add your national and international prefix in the general section.

.

```
.
.
[general]
nationalprefix=0
internationalprefix=00

[interfaces]
msn=54321   # outgoing MSN(s)
incomingmsn=   # incoming MSN(s)
controller=1
softdtmf=0
context=mymenu
.
.
.
```

In the section interfaces you have to configure several things. First of all the outgoing MSNs asterisk is able to make calls with (note this could be more than one).

msn=54321

The MSN on which * will be listening on is specified with the keyword incomingmsn. You can add an * here, to match all MSNs that are called.

incomingmsn=*

controller says which controller * should use.

controller=1

softdtmf is an indicator if we should use softdtmf (1) or not (0)

softdtmf=0

context says which context is called in extensions.conf when a call comes in. Always add the here named context in extensions.conf.

context=isdndefault

Save the file and you are ready to start ...

Just restart * or load the isdn module with:

load chan_capi.so

If everything worked fine you are now able to use your isdn card with asterisk.

Etc....

Configuring Applications

Music On Hold: The moh.conf File

Don't Forget The Timing (Part A)

Voicemail: The voicemail.conf File

Basic SMTP Configuration

Meet-Me: The meetme.conf File

Don't Forget The Timing (Part B)

Sample Configurations

[LOTS OF SAMPLES HERE]

Chapter 4. Scripting and AGI Extensions to Asterisk

AGI In: c, perl, php, etc.

What is AGI?

What languages can I use?

AGI examples in C, perl, php, python, etc.

Chapter 5. Connecting Asterisk to Common VoIP Providers

Overview

Free Service Providers

IAXTEL

Description

Services

What to Expect

Hardware

Setup Examples

Technical Setup

Protocols

Troubleshooting

Help

FWD

Description

Services

What to Expect

Hardware

Setup Examples

Technical Setup

Protocols

Troubleshooting

Help

SipPhone.com

Description

Services

What to Expect

Hardware

Setup Examples

Technical Setup

Protocols

Troubleshooting

Help

XVOIP

Description

Services

What to Expect

Hardware

Setup Examples

Technical Setup

Protocols

Troubleshooting

Help

Commercial Service Providers

NuFone

Description

Services

What to Expect

Hardware

Setup Examples

Technical Setup

Protocols

Troubleshooting

Help

VoicePulse

Description

Services

What to Expect

Hardware

Setup Examples

Technical Setup

Protocols

Troubleshooting

Help

XVOIP

Description

Services

What to Expect

Hardware

Setup Examples

Technical Setup

Protocols

Troubleshooting

Help

Technical Issues

More Information

Chapter 6. Advanced Asterisk Configuration

Agents and the Asterisk ACD

Text-To-Speech: Festival

CLASS Features (John Todd?)

Fax (Software Fax) (Steve Underwood?)

Sphinx Speech Recognition (ASR)

Distributed Asterisk (Clustering/TDMoE)

TDMoE (Time Domain Multiplexing over Ethernet)

To use TDMoE you MUST have a zaptel interface configured somewhere on the network. It can be any zaptel interface, doesn't have to be a E400P, an X100P will do. Why? Timing. Samples. Something like that. Just do it. Ofcourse a dummy ZAP interface like ztdummy or ztrtc might work, but I haven't tried it as yet. If somebody has please do update this.

What is TDMoE?

Well, we all know ethernet right? Its prolly the most popular network infrastructure on Layer2 that the IP world knows. Time-division multiplexing (TDM) puts multiple data streams in a single signal by separating the signal into many segments, each of a short duration (timing). Each individual data stream is reassembled at the receiving end based on this timing.

The circuit that combines signals at the source (transmitting) end of a communications link is known as a multiplexer. It accepts the input from each individual end user, breaks each signal into segments, and assigns the segments to the composite signal in a rotating, repeating sequence. The composite signal thus contains data from multiple senders. At the other end of the long-distance cable, the individual signals are separated out by means of a circuit called a demultiplexer, and routed to the proper end users. A two-way communications circuit requires a multiplexer/demultiplexer at each end of the long-distance, high-bandwidth cable. But, in TDMoE, * serves as the mux/demux. Lets look at the how the configuration is done first, and then do a practical example.

The configuration to define a dynamic span (TDMoX) basically entails FOUR parameters. Look at the sample config from zaptel. Its got an example in it, ala:

```
# Next come the dynamic span definitions, in the form:
# dynamic=<driver>,<address>,<numchans>,<timing>
#
# Where <driver> is the name of the driver (e.g. eth), <address> is the
```

```
# driver specific address (like a MAC for eth), <numchans> is the number
# of channels, and <timing> is a timing priority, like for a normal span.
# use "0" to not use this as a timing source, or prioritize them as
# primary, secondard, etc.  Note that you MUST have a REAL zaptel device
# if you are not using external timing.
#
# dynamic=eth,eth0/00:02:b3:35:43:9c,24,0
```

- First you define the driver (which is eth for ethernet)
- Second is the driver dependent address (REMOTE nic MAC address)
- Third is the number of channels to be configured
- And, lastly, what sort of timing to provide

Timing Notes: 0 for no timing, 1 for primary, 2 for secondary, the difference is that it uses the primary to turn the zaptel gears unless it's in alarm, in which case it will take from the secondary and so on.

The driver is generally "eth" since currently we don't have any other TDMoX drivers, although FireWire would be very very nice. [kram]

The address is **<eth interface>/<macaddress>[/subaddr]**

The sub address is optional, and allows you to define more than one span on a single eth interface / macaddress pair

By configuring this, you end up with a new span, similar to how the T1/E1 spans configured for the E/Tx00P cards. Access to the channels configured above is via /etc/asterisk/zapata.conf.

You can configure signalling and all just as though they were T1's or E1's, so you can run RBS or you can run PRI or whatever, they even generate RED and YELLOW alarm just like real T1's and E1's. We're still debating whether you can run ccs on it.

You do NOT need to configure a specific span=blah,blah in zaptel.conf for this, the dynamic span definition will take care of that.

Remember that TDMoE works at the ethernet layer, all you need to configure is MAC addresses and ethernet interfaces.... so in theory you could TDMoE over 802.11 (low-cost last mile) or cipe (encrypted PRI), the possibilites are limitless (well as limitless as csmacd can get)... IP does not come into play here at all...

-- SIMPLE 2 MINUTE EXAMPLE #1 --

suppose, if i have two * boxen running... lets say merry and pippin...

merry has an X100P in it and a nic, pippin just has a nic

in merry zaptel.conf we have --------

```
fxsls=1                          # this be the X100P
dynamic=eth,eth0/00:D0:B7:89:E3:86,30,0 # put the MAC of pippin nic here
e&m=2-31                         # you can use ANY signalling
```

in pippin zaptel.conf we have --------

```
dynamic=eth,eth0/00:50:FC:65:33:A1,30,1 # note the timing "1", merry's mac
```

```
e&m=1-30                    # same signalling as merry
```

from this point on its like any of the friendly zaptel channels we're already used to....

in merry zapata.conf we have ----------

signalling=em channel=>2-31

in pippin zapata.conf we have ---------

signalling=em channel=>1-30

load the appropriate modules, ztcfg on merry, zttool, you should have RED in the alarms.... and a dynamic span configured (not up, but configured)

do the same on pippin, bingo, the alarms should turn to OK, and you have the zap channels available for use....

```
[root@pippin ~]$ lsmod
Module              Size  Used by    Tainted: P
ztd-eth             4032   0  (autoclean) (unused)
ztdynamic           8544  30  (autoclean) [ztd-eth]
zaptel            177088  60  [ztdynamic]
ppp_generic        27392   0  [zaptel]
slhc                6844   0  [ppp_generic]
```

This listing is with asterisk running, and zapata using the channels. If you got this far, you're good to go.

Enjoy... and mail any samples, suggestions, improvements... always welcome

Hail * !

Todo: multiple ethernet cards (local and remote), other signalling examples, dummy eth driver to loopback test, caveats, benefits of TDMoE, comparision of various signalling, cook dinner

ENUM/E164 Call Routing (LCR)

Databases and Asterisk

PostgreSQL and Applications

CDR and MySQL

AstDB - The built-in database

Chapter 7. Common Issues

Music on Hold/MP3 Playback

Proper Version of MPG123

Timing: zaptel/ztdummy/ztrtc

DTMF over SIP

Inband only works on G.711 ulaw/alaw

SIP-INFO

RFC2833

The "Flash"

Internationalization of Asterisk

Tones and Ringback

Call Supervision

SIP and NAT

Optional/Added Codecs

G.729

G.723

Message Waiting Indication

Common Hardware Device Issues

Grandstream BT100 Series

Cisco ATA-186

Cisco 79XX Series

SNOM VoIP Phones

Carrier Access Channel Banks

Zhone Channel Banks

Echo Cancellation Issues

Interfacing with Legacy PBX Equipment

Nortel Meridian/Norstar

Avaya Definity Systems

Others (Mitel, Aspect, Telrad, Vodavi, Dialogic, etc.)

How to politely use the Asterisk-Users List

How to politely use the Asterisk IRC channel

Chapter 8. Creating Asterisk Applications in C

Appendix A. Sources of Additional Information

Appendix B. Applications Reference

Appendix C. CLI Commands Reference

Appendix D. Manager Commands Reference

Appendix E. The Asterisk C API Reference

Appendix F. Other Open Source Telephony Systems

Glossary of Asterisk & Telecom Terms

FXO

Foreign eXchange Office

When a customer receives phone service from a central office other than the one that would normally serve them, the line between the customer and the "Foreign" office is called a "Foreign Exchange" line and FXS (Foreign eXchange Station) is the station end. FXO (Foreign eXchange Office) is the office end of the line. FXO is also used to refer to the type of interface on phone equipment. An FXO interface receives power and ring signals. An FXS interface provides power and and ring signals.If you want to connect your phone line to your computer so that it can make and answer calls, you need to add an FXO interface to your computer. If you want to connect an ordinary telephone to a computer, you need a card in the computer with an FXS interface.

FXS

Foreign eXchange Station

See FXO

PSTN

Public Switched Telephone Network

i.e. the phone service we use for every ordinary phone call.

ADSI

Analog Display Service Interface

A complex set of standards for the telecom industry. Built off of FSK keying used by CallerID, ADSI is capable of remotely controlling a screenphone with softkeys. Effectively, session based applications can be used when the phone is online, or scripts can be preprogrammed into the phone for when no ADSI connection has been established (on or off hook). Originally, this technology was made by Telcos who thought they would use it to offer services to residential customers. They envisioned such features and buying airplane tickets from your screenphone. Adoption has been mixed. In order to protect their interests, all phones appear to be locked with a programming "code" to prevent users from using one phone provided by one company for a competing service. As such, it can be difficult to get a phone that you can program for yourself (if you have the software). Needless to say, getting codes for an existing phone is nearly impossible (it is incomprehensible to the support staff that anybody but a telco will be programming said phones). Rather, you need to order a phone specifically because you already have the code (eventually you can find a contact that will help you here).

Colophon

This document was written as an excuse to become more familiar with the Docbook format, and to contribute back to the Asterisk project.