



From Technologies to Solutions

AsteriskNOW

A practical guide for deploying and managing an Asterisk-based telephony system using the AsteriskNOW software appliance

Nir Simionovich

[PACKT]
PUBLISHING

AsteriskNOW

A practical guide for deploying and managing
an Asterisk-based telephony system using the
AsteriskNOW software appliance

Nir Simionovich



BIRMINGHAM - MUMBAI

AsteriskNOW

Copyright © 2008 Packt Publishing

All rights reserved. No part of this book may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, without the prior written permission of the publisher, except in the case of brief quotations embedded in critical articles or reviews.

Every effort has been made in the preparation of this book to ensure the accuracy of the information presented. However, the information contained in this book is sold without warranty, either express or implied. Neither the author, Packt Publishing, nor its dealers or distributors will be held liable for any damages caused or alleged to be caused directly or indirectly by this book.

Packt Publishing has endeavored to provide trademark information about all the companies and products mentioned in this book by the appropriate use of capitals. However, Packt Publishing cannot guarantee the accuracy of this information.

First published: March 2008

Production Reference: 1290208

Published by Packt Publishing Ltd.
32 Lincoln Road
Olton
Birmingham, B27 6PA, UK.

ISBN 978-1-847192-88-2

www.packtpub.com

Cover Image by Vinayak Chittar (vinayak.chittar@gmail.com)

Credits

Author

Nir Simionovich

Project Coordinator

Aboli Mendhe

Reviewers

Kimberly Collins

Kristian Kielhofner

Indexer

Hemangini Bari

Acquisition Editor

Viraj Joshi

Proofreader

Chris Smith

Technical Editor

Akshara Aware

Production Coordinator

Shantanu Zagade

Editorial Team Leader

Mithil Kulkarni

Cover Work

Shantanu Zagade

Project Manager

Abhijeet Deobhakta

Foreword

Asterisk® has grown from the very humble beginnings of being my own PBX since I couldn't afford to buy one, and has grown into a world-wide phenomenon, becoming successful due to both the ideas behind it as well as the open-source development model. The usability and usefulness of Asterisk as part of an IP PBX or other telephony system versus a proprietary phone system can be compared in part to the difference between the Betamax and VHS video standards (except, of course that Asterisk is both the open system *and* the best quality system). Betamax, while of an initial high quality, was a proprietary system whose technology could only be advanced by the original creators of the standard. The VHS standard, on the other hand, was made available to a larger development base and thus resulted in more innovation and development. The end result was that the open standard surpassed the proprietary standard in quality, usability, and value. The results are similar as Asterisk has been adopted by a large development community, and resulted in innovation and ease of use that has surpassed traditional technologies.

Every business (and for that matter, pretty much every home) needs a phone system of some level. How to create a system, however, has historically been left to very technical people (even originally in the case of Asterisk). AsteriskNOW™, a software appliance which includes Asterisk as well as the AsteriskGUI™, was created in order to lower the barrier to use and make setting up one's own phone system much less daunting. In a world of GUI-oriented applications, it made sense to create a GUI which could also be useful as well as inspire innovation and creativity.

The AsteriskGUI™, and the code beneath it, is unique compared to other GUIs used in conjunction with Asterisk because the all important Asterisk configuration files can be edited in both the GUI and the command line. Changes made to your IP PBX via the GUI are reflected in the Asterisk configuration files, and vice versa. Thus a novice user, as well as an experienced Asterisk programmer, can use AsteriskNOW in ways that best suit their needs.

Even with AsteriskNOW's ease of use, setting up an IP PBX may not be as easy as it sounds. The book you now hold in your hands is a guide which will assist you in setting up an AsteriskNOW system. If you are new to telephony, you'll gain an understanding of the basic concepts as well. If you are experienced with IP PBX solutions, you'll find information which may help with an AsteriskNOW solution you are developing. The open-source community often provides further assistance for new users on setup, configuration, and creating solutions, and having read this book, you'll get much better support since you've already gotten off to a great start.

Enjoy your experience with Asterisk and AsteriskNOW! And remember to contribute to the ever growing community of Asterisk users and developers who have made it possible for you to create your own PBX, whether it's through code contribution, documentation or just helping other users who are a few steps behind you.

Best Wishes,

Mark Spencer

Original Author and Project Manager, Asterisk CTO, Digium

About the Author

Nir Simionovich has been involved with the open-source community in Israel since 1997. His involvement with the open-source community started back in 1997, when he was a student in the Technion, Israel's Technology Institute in Haifa. Nir quickly became involved in organizing open-source events and promoting usage of Linux and open-source technologies in Israel.

In 1998, Nir started working for an IT consulting company (artNET experts Ltd.), where he introduced Linux-based solutions for enterprises and banks. By 2000, Nir had become a SAIR/GNU-certified Linux trainer and Administrator, slowly educating the future generations of Linux admins.

In 2001, Nir moved to the cellular content market, working for a mobile content delivery company (m-Wise Inc. – OTC.BB: MWIS.OB). During his commission at m-Wise, Nir successfully migrated a company that was built purely on Windows 2000 and ColdFusion to open-source technologies, such as Mandrake Linux (today Mandriva), Apache Tomcat, and Kannel (open-source SMS/WAP gateway).

By 2006, Nir had co-founded Atelis (Atelis PLC – AIM: ATEL). Atelis is a Digium distributor and integrator. During the course of 2006, Nir developed an Asterisk-based international operator services platform for Bezeq International, which had replaced a Nortel DMS-300 switch. This platform is currently in use by Bezeq International in Israel, serving over 4000 customers a day.

In mid 2007, Nir left Atelis to become a freelance Asterisk promoter and consultant. Nir currently provides Asterisk consulting and development services to various companies, ranging from early-stage start-up companies, through VoIP service providers and VoIP equipment vendors. In his spare time, Nir is the founder of the Israeli Asterisk users group, the website maintainer of the group and an Asterisk developer, dealing mainly with the localization aspects of Asterisk to Israel.

Nir can be reached at nirs@greenfieldtech.net or through his website <http://www.greenfieldtech.net>.

I believe the first time I ever used Asterisk™ was mid 2002. Back then I was working as the IT Director of a start-up company dealing mostly in the mobile market. Our office PBX was a Panasonic PBX, which used to stop working right when we needed it the most. I was frustrated: the PBX in the office never works right and the PBX technicians that come to fix it never do their job right. Being involved in the open-source community since early 1995, I asked myself: "Isn't there an open-source alternative to this?" —So, I started searching.

I discovered a few projects, but none were really a complete solution besides a solution that was called Asterisk™, from a company in Huntsville called Linux Support Services. I downloaded and installed it, and immediately realized the following: no way would my company migrate from the Panasonic to Asterisk™ at that point in time. So, I started, learned and understood it and waited for my chance.

Approximately six months later, the company had got involved in an SMS-based Callback solution. The initial solution was based on a Cisco AS5300 gateway, which was outsourced from another company for the duration of the development. Once the development had finalized, the company wanted to start the service based on the Cisco equipment only to realize that the cost of building the system would never sustain the projected business model. At that point, I saw the opportunity to take Asterisk and adapt the code base to use Asterisk instead of using a Cisco gateway. I took it up to modify the code along with another programmer. The development and modifications lasted about four weeks, and we got the same functionality using Asterisk — the date was early 2003. The new development was able to sustain the business model, which then evolved into a fully operational SMS callback service.

Since then, I've developed various platforms based upon the Asterisk open-source project. I've established the Israeli Asterisk™ users group community, held the first Israeli Asterisk™ convention, and most importantly, was co-founder of Atelis plc, which is now traded on the London stock exchange (AIM: ATEL). I recently left Atelis plc and established my own small Linux™ and Asterisk™ consultancy firm, which renders consulting services to various Asterisk™-based service companies and Asterisk™-enabled vendors in Israel and around the world.

I would like to take this opportunity to thank some people:

First of all, I'd like to thank my wife for putting up with my rants and raves about Open Source, Asterisk, the amount of hardware and mess on my desk and my complete disregard to anything in the house. Nili, I love you.

To my parents, for putting up with my craziness over the years and the endless nights of me tapping at the console when I was growing up.

To Mark Spencer, for developing Asterisk™ and for creating one of the most innovative tools on the market today. And most importantly, thank you for your help back in 2003, when I needed to install the first BRI interface and had no idea what I was doing in there – Mark was back then sitting in the IRC channel, and was one of the biggest helps to me.

To Schuyler Deerman, who actually connected me with Packt for publishing this book. Schuyler is one of Digium's field marketing person and had become a close friend over the course of our mutual work. Schuyler is currently studying in France.

To Optimus, my first Asterisk™ server, which had suffered and suffered and suffered, till I got it to work as I wanted it. Optimus is currently resting in pieces somewhere down the pile of servers I have at home.

About the Reviewers

Kimberly Collins is a California transplant who found her home in Austin, TX. She has worked in the field of Information Technology and communications for over ten years.

She spent the last two years working for one of the largest hosting companies in the world, and currently is one of their lead administrators and developers of their global VOIP infrastructure.

Occasionally you might catch her in IRC as jgoddess, but if you happen to miss her then you can find her on AIM as womkim or MSN messenger as tattletailes@hotmail.com. You can email her at womkim@gmail.com.

Kristian Kielhofner is VP, Systems Engineering for Star2Star Communications, developer of an end-to-end VoIP architecture. Kristian is responsible for the design and implementation of Star2Star's VoIP services. He is also the creator and lead developer of AstLinux, an embedded Linux distribution for voice and networking appliances. In addition to working on AstLinux and the Star2Star Architecture, Kristian enjoys traveling to speak about free software at events around the globe.

Table of Contents

Preface	1
Chapter 1: An Introduction to Telephony and Asterisk	7
The Basics of Traditional Telephony	8
Circuit Switching	8
A Circuit-Switched Network	9
Signalling System # 7 (SS7)	10
Integrated Services Digital Network (ISDN)	11
The Basics of Voice over IP (VoIP) Technology	13
Session Initiation Protocol—SIP	14
Inter-Asterisk eXchange Protocol—IAX	16
NAT/PAT: IAX2 versus SIP and H.323	17
CODECS—Voice Coder Decoder	17
Asterisk—The Open-Source PBX	19
Asterisk is Dually Licensed—What Does it Mean?	20
Enter the Asterisk—the Future is Here	20
AsteriskNOW—The Asterisk Software Appliance	21
Summary	22
Chapter 2: Building a PBX	23
Objective—Building an Office PBX	23
Physical Connectivity	24
Installation Procedure Outline	25
Downloading AsteriskNOW	26
AsteriskNOW Hardware Requirements	26
The Installation Process	29
Anatomy of the AsteriskNOW Configuration GUI	46
Introduction to the rPath Appliance GUI	47
Summary	48

Chapter 3: Extensions, Phones, and Others	49
An IP Phone is a Simplified Computer	49
AsteriskNOW Extension Management GUI	50
The User Extensions Configuration Options	52
The "User Extension" Configuration Flags	52
The LinkSys 941	55
CounterPath X-Lite—The Worlds Most Popular Soft Phone	60
Summary	63
Chapter 4: Service Providers—Your Connection to the World	65
VoIP Carriers	65
Direct Inward Dialing (DID/DDI) Carriers	66
IP Call Termination Carriers	66
Refilers and Grey Routes	67
PSTN Carriers—Traditional Telephony Providers	68
Configuring an IP Termination Service Provider	69
VoIP Service Providers in AsteriskNOW	69
VoIP Service Providers—Few Examples	72
Inbound DID/DDI Service Providers	72
Termination and Residential Service Providers	73
Connecting to a Custom VoIP Termination Provider	74
Summary	76
Chapter 5: Tentacles of the PBX—The Calling Rules Tables	77
Managing Routing Rules with AsteriskNOW	78
Manually Editing Dial-Plan Logic	79
Summary	84
Chapter 6: "Let me in!"—Inbound Call Routing	85
Inbound DID Routing versus Analog Physical Routing	85
Inbound Routing via DID Numbers	85
Inbound Routing via Physical Ports	86
Routing Type Comparison Table	86
Inbound Call Routing with AsteriskNOW	86
Example 1: Routing in a Single DID Number	88
Example 2: Routing in a Range of DID Numbers	89
Inbound Call Routing in extensions.conf	89
Summary	91
Chapter 7: "For Annoyance, Press 1"—Voice Menus and IVR	93
Four Rules of IVR	93
Voice Menus—AsteriskNOW's IVR Generator	94
Voice Menu Steps—The Voice Menu Flow	95
DISA—Direct Inward System Access	96
Recordings—Menus and System Playbacks	97

Time Based Rules	99
Ring Groups	101
Enough Theory, Back to Voice Menus	103
Summary	107
Chapter 8: Voicemail, Conferencing, and Parking—Advanced PBX Services	109
Comedian Mail—The Asterisk Voicemail System	109
Voicemail General Options	110
Voicemail Message Options	111
Voicemail Playback Options	112
MeetMe Conferencing	112
Conference User and Administrator Key Presses	113
Defining a New Conference Room	114
General Conference Options	114
Conference Password Settings	115
Conference Room Options	115
Call Parking	116
Summary	117
Chapter 9: "Please hold, we'll be with you shortly"—Simple Call Queues	119
Queue General Options	120
Queue Options	121
Utilizing Call Queues	121
Summary	123
Chapter 10: General AsteriskNOW Management—Monitoring, Backups, and More	125
AsteriskNOW General Options	125
Local Extension Settings	126
Agent Login Settings	127
Extension Options	127
AsteriskNOW Backup	128
Asterisk Logs	129
AsteriskNOW System Info	130
AsteriskNOW Active Channels	132
AsteriskNOW Graphs	133
Summary	134
Chapter 11: Hard Core AsteriskNOW	135
AsteriskNOW Advanced Options	135
Music on Hold	136
Why Do You Need Multiple Music On Hold Classes?	137

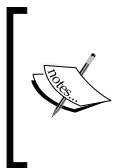
VM Email Settings	137
Global SIP Settings	138
General SIP Settings	139
Type of Service Settings	139
NAT Support Settings	140
Global IAX Settings	141
General IAX Settings	141
Jitter Buffer Settings	141
IAX Registration Options	142
Codecs Settings	143
Change Password	143
Setup Wizard	143
Gaining Root Access to Your AsteriskNOW via SSH	143
The Asterisk Command-Line Interface (CLI)	146
The Asterisk Dial-Plan Language (extensions.conf)	155
Configuration File Structure	156
Extension Pattern Matching	158
Special Extensions in extensions.conf	158
The Asterisk Configuration Directory	159
Summary	160
Chapter 12: Where to from Here?	161
Beyond the Dial Plan—Asterisk Gateway Interface (AGI)	161
AGI Execution Environment	162
AGI Example in PHP	163
AGI Programming API Functions	164
AGI Programming Libraries	166
Asterisk Manager Interface (AMI)	167
Asynchronous JavaScript Asterisk Manager (AJAM)	169
Ideas and Mesh-ups	169
Voice-Enabled Network Monitoring	169
Voice-Enabled Intrusion Detection	170
Voice-Enabled Attendance Clock and Proximity	170
DUNDi—Distributed Universal Number Discovery	170
Summary	172
Appendix A: Jargon Buster	173
Appendix B: Free World Dialup (FWD)	175
Appendix C: AsteriskNOW for Service Providers	179
Index	181

Preface

AsteriskNOW is an open-source software appliance from Digium: a customized Linux distribution, which includes Asterisk (the leading open-source telephony engine and tool kit), the AsteriskGUI, and all the other software needed for an Asterisk telephony system.

This book discusses the installation and configuration of the AsteriskNOW open-source PBX appliance distribution and is written in the form of a self-study guide or a quick cookbook, to get you up and running with AsteriskNOW as fast as possible.

While Asterisk, the open-source PBX is a fairly broad subject to cover – the AsteriskNOW distribution takes the spikes out of installing and using Asterisk, and lowers the bar to the level of an intermediate system's administrator.



This book is based upon AsteriskNOW Beta 6. By the time this book is published, the version of AsteriskNOW may have changed, and new features may have been added to it. This book will enable your descent into the Asterisk world and AsteriskNOW in particular giving you the basics of Asterisk and AsteriskNOW – no matter what version you may use.

What This Book Covers

Chapter 1 introduces the basic concepts of a telephony system, both traditional and IP telephony. The chapter serves as trip down telephony memory lane, explaining the various interfaces, technologies, and terms commonly used in the telephony and telecommunications industry.

Chapter 2 introduces the various hardware elements required for installing your AsteriskNOW PBX system and the AsteriskNOW installation procedure. Pay close attention to the hardware mentioned in this chapter; familiarity with the Digium line of interface cards will make your deployment much easier, when trying to decide which hardware to use.

Chapter 3 deals with the various aspects of configuring extensions and IP phones, the basic elements of an IP telephony system. You will be introduced to two specific types of IP phones – a hardware IP phone (LinkSys SPA-941) and a software IP phone (CounterPath X-Lite).

Chapter 4 deals with the concept of telephony service providers. These are usually your local PSTN providers. In addition, the chapter deals with the concept of IP telephony providers: inbound providers and termination providers.

Chapter 5 explains what routing rules are and how they are processed within the AsteriskNOW operational model.

Chapter 6: Routing calls into and out from your PBX system can be complex. This chapter deals with the various logics that need configuration in order to enable proper call traversal to and from your PBX system.

Chapter 7: Interactive Voice Response and Auto Attendants are corner stones of the PBX market. AsteriskNOW provides a highly versatile and simple interface for configuring and controlling these two elements. This chapter deals with the configuration of an IVR/Auto-Attendant, and most importantly, the rules for building a proper IVR/Auto-Attendant.

Chapter 8 deals with some of the more advanced features of AsteriskNOW. Voicemail, conferencing, and call parking are utilized on a day-to-day basis in every PBX system – pay attention to the voicemail-to-email feature; it may lower your expenses on calling the voicemail system, when you are outside the office.

Chapter 9 deals with configuring call-queues and building a mini call center. While AsteriskNOW is fully capable of serving over 100 agents, this chapter will explain how to create a miniature call center and the concept of skill-based routing.

Chapter 10 takes a look into the general aspects of managing your AsteriskNOW installation, beyond the telephony portion. Like any other computer-enabled service your AsteriskNOW system will require maintenance such as backups, monitoring, and more.

Chapter 11 is meant for the hard-core user looking to do more than what the GUI interface has to offer. This chapter should be approached with care; if you are not an experienced Asterisk/Linux user or a developer looking to develop applications for AsteriskNOW you could skip this chapter.

Chapter 12 is meant as a short look ahead to other possibilities enclosed with your PBX. AsteriskNOW and Asterisk are not only a PBX, but actually a rich telephony development platform, capable of doing much more than being a PBX.

Appendix A is a jargon buster.

Appendix B takes a quick look at how to configure Free World Dialup (FWD) services for your AsteriskNOW PBX system. If you have multiple offices, utilizing FWD to interconnect freely between them will enable cost savings on inter-office communications.

Appendix C shows how a service provider can modify the AsteriskNOW distribution to add their own service provider entry directly into the AsteriskNOW GUI.

What You Need for This Book

This book is a practical guide to get you up and running with AsteriskNOW. In order to install a fully working PBX system using AsteriskNOW, you will need the following:

- A PC to install the AsteriskNOW software appliance. Hardware requirements for this PC are indicted in Chapter 2.
- A Windows, Linux or MAC based PC to install the counter-path X-Lite soft phone.
- A Digium TDM400 card equipped with 3xFXO modules and 1xFXS module. This card can be purchased online at <http://www.voipsupply.com>.
- A LinkSys 941 IP Hardware phone (optional) available at <http://www.voipsupply.com>.

Conventions

In this book, you will find a number of styles of text that distinguish between different kinds of information. Here are some examples of these styles, and an explanation of their meaning.

Code words in text are shown as follows: "The interesting portion of the above line is the `_9xxx!`"

A block of code will be set as follows:

```
; Extension names may be numbers, letters, or combinations
; thereof. If an extension name is prefixed by a '_'
; character, it is interpreted as a pattern rather than a
; literal. In patterns, some characters have special meanings:
;
; X - any digit from 0-9
; Z - any digit from 1-9
```

New terms and **important words** are introduced in a bold-type font. Words that you see on the screen, in menus or dialog boxes for example, appear in our text like this: "In the above network diagram, every **Central Office Exchange** is connected to the other exchanges".



Important notes appear in a box like this.



Tips and tricks appear like this.

Reader Feedback

Feedback from our readers is always welcome. Let us know what you think about this book, what you liked or may have disliked. Reader feedback is important for us to develop titles that you really get the most out of.

To send us general feedback, simply drop an email to feedback@packtpub.com, making sure to mention the book title in the subject of your message.

If there is a book that you need and would like to see us publish, please send us a note in the **SUGGEST A TITLE** form on www.packtpub.com or email suggest@packtpub.com.

If there is a topic that you have expertise in and you are interested in either writing or contributing to a book, see our author guide on www.packtpub.com/authors.

Customer Support

Now that you are the proud owner of a Packt book, we have a number of things to help you to get the most from your purchase.

Errata

Although we have taken every care to ensure the accuracy of our contents, mistakes do happen. If you find a mistake in one of our books – maybe a mistake in text or code – we would be grateful if you would report this to us. By doing this you can save other readers from frustration, and help to improve subsequent versions of this book. If you find any errata, report them by visiting <http://www.packtpub.com/support>, selecting your book, clicking on the **Submit Errata** link, and entering the details of your errata. Once your errata are verified, your submission will be accepted and the errata added to the list of existing errata. The existing errata can be viewed by selecting your title from <http://www.packtpub.com/support>.

Questions

You can contact us at questions@packtpub.com if you are having a problem with some aspect of the book, and we will do our best to address it.

If you run into issues while using this book, feel free to logon to AsteriskNOW support forum on the author's website located at <http://www.greenfieldtech.net/support/>.

1

An Introduction to Telephony and Asterisk

Telephone, n. An invention of the devil which abrogates some of the advantages of making a disagreeable person keep his distance. — Ambrose Bierce.

What is a telephone? While Ambrose Bierce refers to it as an invention of the devil, for most of us a telephone is nothing more than a communication device. For some a telephone is a complex electronic device they cannot live without (have you ever seen a stock broker work?), while for others, it is simply an innovative piece of equipment in their office or home. Telephones exist in many shapes and sizes, but one thing unites them all — their functionality. Be it the most complex phone in the world or the silliest kiddy phone, they essentially enable us to communicate with other people.

Unlike other technologies, telephony is considered highly simplistic from the user's point of view. This chapter imparts the basic knowledge necessary to start your journey into the world of telephony. By the end of this chapter, you will have learned:

- The basics of telephony — The Central Office Exchange (COX) and the Private Branch Exchange (PBX).
- The basics of telephony interfaces and telephony wiring — FXO, FXS, BRI, and PRI.

Once you have acquired the above mentioned knowledge, continue your journey into the world of Asterisk — the open-source PBX, and AsteriskNOW — the Asterisk appliance distribution.

The Basics of Traditional Telephony

One of the most common terms in telephony is the **exchange** or **switch**, which refers to the actual device or mechanism that connects the parties who wish to converse over a telephone line. In most cases, the exchanges are located in a centralized location, interconnected with hundreds, sometimes thousands of end terminals (telephones). Exchanges are then interconnected among themselves via connecting trunks. Trunks are connections capable of carrying multiple phone calls concurrently, thus enabling calls to traverse among exchanges.

One might ask, "If there are thousands of end terminals and a multitude of exchanges within a network, how does a call know where to ring?" To understand this, take a deeper look into the term **circuit switching**.

Circuit Switching

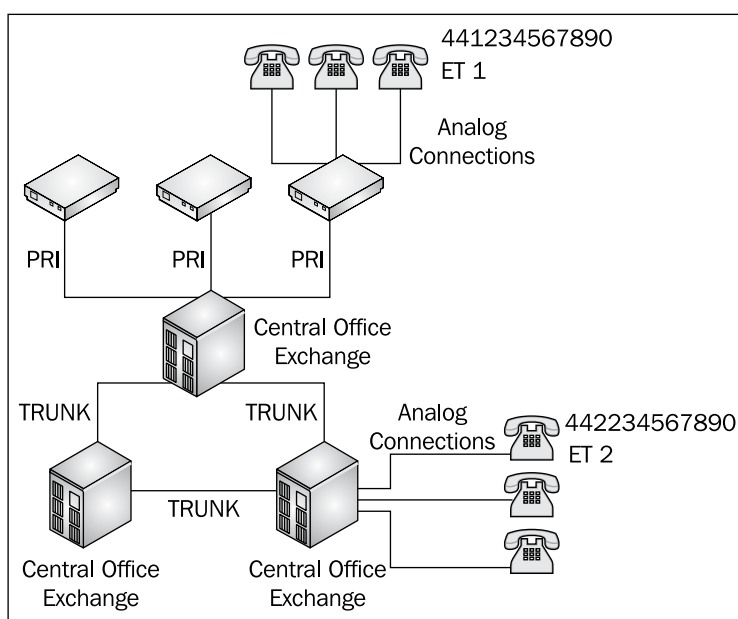
Circuit switching refers to the methodology of interconnecting end terminals before actual information can traverse between them. You may think of circuit switching as pre-defining the route of a train, before the train actually leaves the station. Once a circuit is established, it is dedicated and can't be interrupted, until the circuit is released by one of the end terminals.

The early day telephone exchange was a pure circuit-switched network. In fact, the circuit-switching device was actually a human operator who connected wires from one side of the circuit board to the other. Some places in the world still use these old telephone exchanges, mainly in situations when automatic circuit switching isn't available. In fact, some armies in the world still use manual switching boards for field communications — amazing, isn't it?



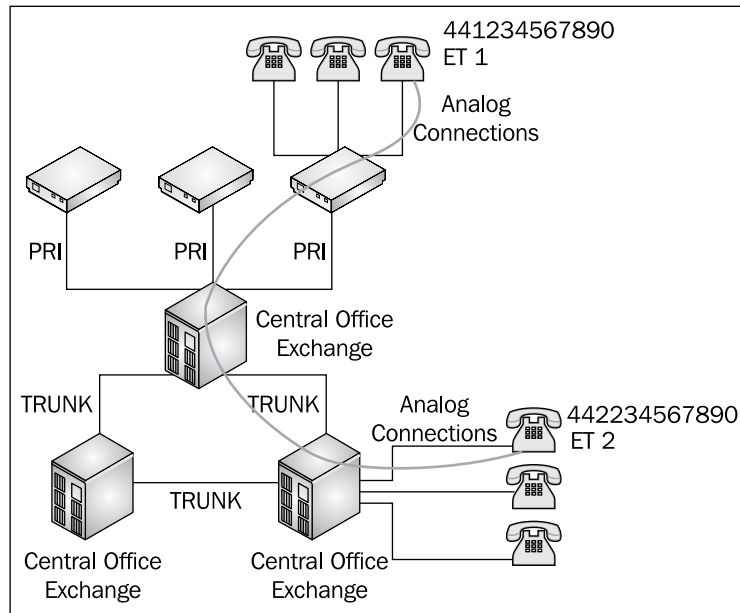
A Circuit-Switched Network

A circuit-switched network is usually made up of the following elements – Central Office Exchanges, Private Branch Exchanges, and end terminals. Examine the following diagram:



In the above network diagram, every **Central Office Exchange** is connected to the other exchanges. The following diagram can either describe the entire network, or a small portion of the network, illustrating a single town telephony network. The end terminals ET1 and ET2 have been assigned numbers **441234567890** and **442234567890** respectively. Once a call is initiated from ET1 to ET2, a dedicated link (circuit) is established across the entire network, connecting the two end terminals together.

In the following diagram the currently allocated circuit from ET1 to ET2 is shown.



In the above diagram observe that the connection between ET1 and ET2 prevents other terminals from establishing a circuit to ET1 or ET2. At the same time, other end terminals are fully capable of establishing circuits among themselves, even while traversing the same trunks as the circuit from ET1 to ET2.

The trunks connecting the Central Office Exchanges would usually be of type SS7, connections from the Central Office Exchange to a Private Branch Exchange would usually be of types PRI, BRI, or FXO, and the connection from the Central Office Exchange to an analog end terminal would always be of type FXO. Each of these connectivity methods is discussed in the following sections.

Signalling System # 7 (SS7)

While SS7 is a common trunk when interconnecting Central Office Exchanges, its usage in the Asterisk environment is still highly experimental. The various aspects of the SS7 signalling protocol are beyond the scope of this book; however, some preliminary information is included. Unlike traditional analog signalling systems, SS7 separates the signalling channel from the rest of the media channels. Signalling refers to the exchange of information between call components required to provide and maintain service, mainly Central Office Exchanges. If at any time the signalling of a currently allocated call becomes non-existent, the exchanges remove the associated media connection between the interconnected call components.

Integrated Services Digital Network (ISDN)

While ISDN utilizes the same methodologies as a circuit-switched network, to provide better voice and data exchange between end terminals, it utilizes the same infrastructure as a regular telephony network. ISDN consists of two separate connectivity interfaces – Basic Rate Interface (BRI) and Primary Rate Interface (PRI).

ISDN Basic Rate Interface (BRI)

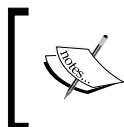
An ISDN BRI connection consists of two bearer channels (B-channels), each one capable of carrying a maximum throughput of 64kbps, and a single data channel (D-channel) of 16kbps. B-Channels carry either voice or data, while the D-Channel carries signalling information. When utilized in voice mode, a BRI can carry two phone calls at the same time. When utilized in data mode, a BRI would be able to combine the two channels into a single data path of 128kbps. While ISDN BRI is highly common in Europe, its usage in USA is very rare.

ISDN Primary Rate Interface (PRI)

ISDN PRI connections are separated into two type – E1 and T1. While E1 circuits are mainly used in Europe, Africa, and Asia, the T1 interface is mainly used in North America and Japan. The differences between the two are as follows:

ISDN Type	D-Channels	B-Channels	Aggregate Throughput
E1 PRI	1 (normally channel 24)	30	$30 * 64\text{kbps} = 1920\text{kbps} + 16\text{kbps} = 1936\text{kbps}$
T1 PRI	1 (normally channel 16)	23	$23 * 64\text{kbps} = 1472\text{kbps} + 16\text{kbps} = 1488\text{kbps}$

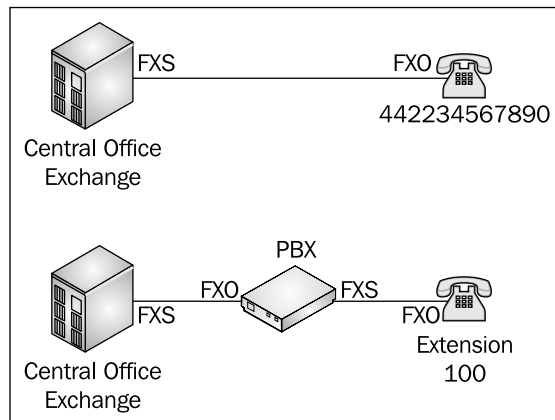
In many situations, an E1 PRI circuit will be referred to as a 2Mbps circuit, while a T1 circuit will be referred to as a 1.5Mbps circuit. While a PRI is much larger than a BRI interface, in terms of size, both utilize the same signalling operational methodologies, and also enjoy a similar feature set.



To learn more about the ISDN protocol, visit the ISDN information page on Wikipedia, which has a multitude of information and is fairly accurate for the novice user. ISDN on Wikipedia: http://en.wikipedia.org/wiki/Integrated_Services_Digital_Network.

Foreign Exchange Office (FXO) and Foreign Exchange Station (FXS)

These two interfaces tend to confuse many people, and the confusion is very much understandable. To make it simpler, observe the following diagram, describing the location of each of these interfaces.



On examining the above diagram, one would immediately notice that the FXO interface is located at the receiving end of the connection, while the FXS interface is the one originating the service. This means that one cannot exist without the other; to work an FXO interface on one side requires an FXS interface on the other and vice versa.

Another way of thinking about it would be that analog telephony requires the generation of on-hook and off-hook signalling to the network generated by an FXO interface. Thus, the FXO interface would be connected to a device capable of generating these signals – an analog phone device or the incoming port of a PBX. The FXS interface would be the one capable of reading these signals, a device capable of generating tones and voltage – thus the connection at the Central Office Exchange or the PBX outgoing extension port.

While Asterisk is capable of handling other types of signalling interfaces, the above-mentioned interfaces are the most common interfaces that you will ever use.

[ To learn more about FXS and FXO interface, please visit the
voip-info.org website.
voip-info.org on FXO:
<http://www.voip-info.org/wiki/view/FXO>
voip-info.org on FXS:
<http://www.voip-info.org/wiki/view/FXS>]

The Basics of Voice over IP (VoIP) Technology

Having its roots planted back in early 1995 by the world's first VoIP Company, VocalTec, VoIP has enjoyed a rapid growth in usage, adaptation, and recognition. To understand how VoIP has evolved over the years, it is important that you take a small walk down memory lane.

- **1995:** Vocaltec, a start-up company based in Israel, released the first ever PC-to-PC VoIP application called "Internet Phone" based upon the H.323 signalling protocol and simplistic codecs. Due to the nature of the Internet in 1995 and the lack of proper broadband connectivity, the Internet Phone application made waves, but not the big splash it was supposed to make.
- **1998:** First adaptations of PC-to-Phone connectivity. VoIP had started migrating into the carrier environment, with first adaptations in the US. Reports indicated that by the end of 1998, 1% of the US telephone traffic was based on VoIP technologies.
- **1999 till 2000:** VoIP usage grew stronger within the telecommunication market lowering the cost of international call termination (call termination refers to the action of transmitting a telephone call over any medium to the terminal receiving the call). The first version of Asterisk, the open-source PBX, was released to the open-source community. While H.323 was still mostly used, early adaptations of SIP-based signalling started to show.
- **2001 till 2004:** SIP-based service providers and SIP-based interconnectivity slowly replaced old H.323-based services. Asterisk was rapidly adopted by early adopters, who modified and moulded it to their needs.
- **2005:** The first stable release of Asterisk was released to the public. VoIP-based telephony services were becoming dominant, with Vonage leading the market in the US.
- **2006:** Version 1.2 of Asterisk was launched, which brought greater stability and new features to the project. Asterisk had been slowly disrupting the market, up to a point where the industry considered Asterisk as a replacement platform for traditional class-5 applications.
- **2007:** Version 1.4 of Asterisk was launched, which clearly marked the entry of Asterisk into the telecoms mainstream.

Asterisk as a project supports a multitude of VoIP signalling protocols, such as H.323, SIP, IAX2, and MGCP. A primer to the VoIP technologies that are the most common in AsteriskNOW – SIP and IAX2 is presented in the following sections.

Session Initiation Protocol—SIP

While H.323 conforms to the ISDN Q.931 signalling and control protocol, SIP conforms to HTTP as a signalling and control protocol. Confusing isn't it?

In other words, H.323 utilizes Q.931 as its signalling and control protocol, a binary stream-based protocol, identical to the one used in the ISDN standard.

Question: Why would an IP protocol utilize a signalling methodology of an electronic nature, instead of defining a new signalling methodology?

Answer: Well, the answer to that isn't clear, but when examining the people involved in the creation of the H.323 standard, it makes sense. H.323 is an umbrella recommendation from the ITU-T, the telecommunications standards organization. This means that the ITU-T engineers simply adopted what they already knew to IP. The result, while being utilized widely, was a highly complex signalling layer, of binary nature; and which required a highly skilful engineer to perform debugging and configuration tasks.



To learn more about the H.323 signalling protocol, please visit the following:

H.323 on Wikipedia:

<http://en.wikipedia.org/wiki/H323>

H.323 on voip-info.org:

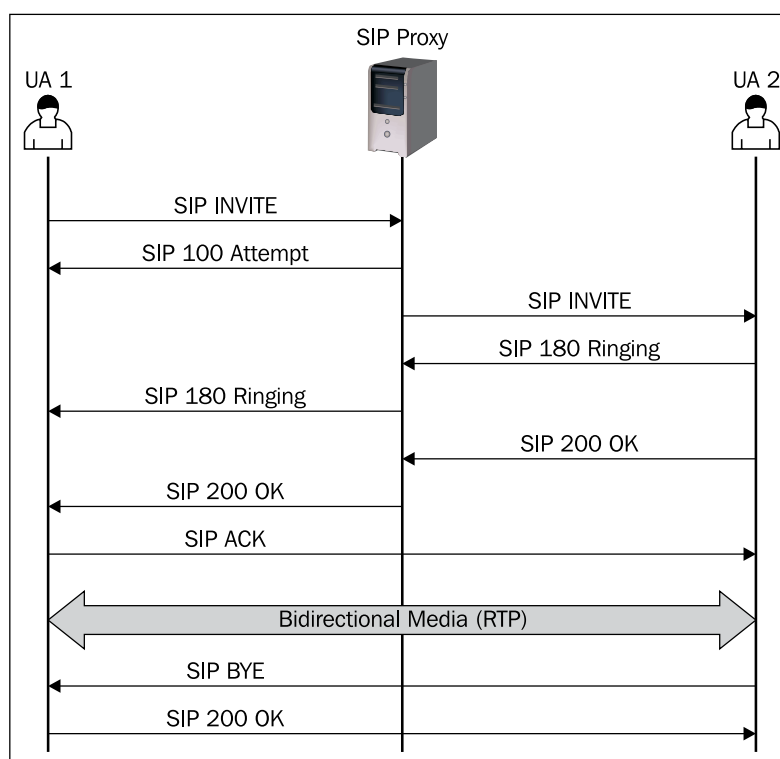
<http://www.voip-info.org/wiki/view/H.323>

As the technologies around the evolution of the Internet evolved, it was clear that a new VoIP protocol would be devised: one that would conform to the workings on the Internet, and be designed from the ground up, based upon Internet technologies and methodologies. The people at the IETF (Internet Engineering Task Force) drafted the Session Initiation Protocol, other wise known as SIP. Unlike H.323, SIP utilizes text-based signalling, very much similar to the HTTP protocol. The signalling is performed utilizing a preset number of signalling and control messages, while media is traversed utilizing a standardized media transfer layer (RTP will be discussed later). In addition to the above, the IETF engineers had decided that SIP would not only be a voice only signalling layer, but would also enable the signalling for any other session-oriented applications. These days you can find Instant Message applications and Video Conferencing solutions based upon the SIP signalling protocol.

According to the IETF standard (RFC 3261), every SIP-compatible application (software or hardware) **must** conform to the following SIP methods:

Method	Explanation
INVITE	Invite another User Agent (UA) to a session.
RE-INVITE	Change a running session.
REGISTER	Register a location with a SIP Registrar server, typically a SIP Proxy server or an IP-enabled PBX system.
ACK	Used to facilitate reliable message exchange for INVITE.
CANCEL	Cancel an invite.
BYE	Terminate a session.
OPTIONS	The SIP OPTIONS method allows a UA to query another UA or a proxy server as to its various capabilities. The discovered capabilities may include codec types, payload types, and other UA and proxy capabilities.

In addition to the above methods, various RFC documents have expanded this method list to add new methods, some with generic functionalities, and some with vendor-specific functionalities. To better understand the way SIP works, examine the following SIP call scenario.



As seen from the preceding diagram, every SIP-based session starts with an **INVITE** method. The **INVITE** is a request to the **SIP Proxy**, which indicates that **UA 1** wishes to talk to **UA 2**. The **SIP Proxy** will respond with a result code of **100**, indicating that it is currently trying to locate and connect the requested UA. The **SIP Proxy** then proceeds to send another **INVITE** to **UA 2**, which responds with result code **180**, indicating that the UA is currently ringing the call. If UA 2 had been on a call, the returned result code would have been **486**, indicating that the UA was currently busy. Once the SIP session has been established, a bidirectional RTP stream is constructed between the UAs. This means that while SIP signalling would traverse the SIP proxy server, the actual media (voice/video/other) would communicate peer-to-peer between the nodes.



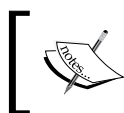
To learn more visit:
RFC 3261 – The Session Initiation Protocol RFC
<http://www.ietf.org/rfc/rfc3261.txt>

Inter-Asterisk eXchange Protocol—IAX

First things first, IAX is pronounced as "eeks" and not "eye-ay-ex". Why is this important? Well, when you will attend your first Asterisk convention (or hopefully, if you are the organizer of such event), it is important that your fellow Asteriskians will distinguish you from a newbie user.

IAX was developed by Digium for the sole purpose of interconnecting between Asterisk servers—hence the name. While most VoIP signalling protocols perform a strong separation between signalling and media, IAX doesn't perform that separation. As a result, IAX is able to traverse signalling and media over a single User Datagram Protocol (UDP) port, while H.323 and SIP require a single UDP port for signalling and multiple dynamic UDP ports for media.

An added value is the ability to trunk multiple call sessions into a single dataflow. This feature preserves bandwidth and network resources when interconnecting Asterisk servers, utilizing the IAX protocol. Another advantage is security, as IAX is capable of performing authentication using plain text, MD5 hashing, and RSA key exchanges, making it more secure than Asterisk's SIP implementation. As Asterisk progresses, we will surely see advancements in that field too, most probably adding TLS functionality in addition to the above.



AsteriskNOW is a closed distribution thus; the OpenVPN packages are not installed on it by default.

NAT/PAT: IAX2 versus SIP and H.323

Network Address Translation (NAT) and Port Address Translation (PAT) pose a serious problem for traditional VoIP signalling protocols such as SIP and H.323. In the previous diagram, imagine that NAT/PAT devices block the access from **UA 1** to **UA 2**. That means that the bidirectional RTP stream won't traverse the network correctly.

Question: Why won't the RTP stream traverse correctly?

Answer: Both SIP and H.323 carry a portion of the IP information of the UA. This means that if a UA is located behind a NAT/PAT network, essentially, it is located within a private IP address space, which is non-routable in many cases. This means that while the signalling may pass correctly, the RTP is "trying" to establish itself between IP addresses that can't connect to one another – no RTP can pass between the UA's in such a situation.

While both H.323 and SIP provide various solutions to solve the NAT/PAT issue, none of them solve the issue completely – and in some cases, are unable to solve the problem at all. In addition, in highly complex enterprise environments, when several NAT/PAT networks may be cascaded (this is an extreme über-sysadmin case; however, it may happen), rather all these solutions will fail.

IAX2 is different, thanks to IAX's single port implementation; it is able to traverse NAT/PAT devices easily. The only thing that would be required is to allow access on the IAX2 UDP port (4569) and voice would traverse the network easily.

As you can see, in terms of security, network simplicity, and bandwidth IAX2 is the best choice for interconnecting Asterisk-based servers (be it AsteriskNOW or any other Asterisk-based product) than SIP and H.323.

CODECS—Voice Coder Decoder

Without diving too deep to the math, a codec is nothing more than a mathematical model, capable of sampling voice streams and compressing them.

An ISDN BRI line is capable of traversing two telephone calls, running on each of its B-channels. This means that in normal PSTN telephony, the codec used to traverse voice requires a throughput of 64kbps.

Now, imagine that you were able to take that 64kbps and utilize it differently – say that you can compress your 64kbps voice stream into something smaller. That is exactly the purpose of the codec – to enable this compression. While a codec may conserve bandwidth, it will consume computing power – like any other piece of software, a codec is a mathematical model that needs to be implemented. Some codecs carry a high computing power toll while others carry a small one; it all depends on your usage and bandwidth. There is no right or wrong codec to use; it all depends on the use.

Your choice of codecs has a direct impact on the performance of your PBX system. Utilization of high computing codecs (such as G.729A and iLBC) will lower the number of concurrent calls on your PBX.

Various codecs exist, each one with its own strengths and weaknesses. The following table summarizes Asterisk and AsteriskNOW's codec compatibility and availability.

Codec	Bitrate (kbps)	Passthrough	Transcoding	Default Availability in Asterisk
G.711 (*)	64kbps (US: u-law, Others: a-law)	Fully Supported	Fully Supported	Yes
G.726 (*)	16, 24 or 32kbps	Fully Supported	Fully Supported	Yes
G.723.1 (**)	5.3kbps or 6.3kbps	Fully Supported	Requires License	Digium Hardware
G.729A (**)	8kbps	Fully Supported	Requires License	Digium Hardware Digium Software
GSM (*)	13kbps	Fully Supported	Fully Supported	Yes
iLBC (**)	13.3kbps or 15.2kbps	Fully Supported	Fully Supported	Yes
Speex (**)	Variable (2.15kbps – 22.4 kbps)	Fully Supported	Fully Supported	Yes (Although, requires additional third-party libraries and additional compilation in most cases)



(*) This codec is considered a low computing power consumer.

(**) This codec is considered a high computing power consumer.

While Asterisk defaults to several codecs, it is clear that better codecs require a license of some sort. Codecs, like many other algorithms, are patent protected, each one belonging to a different business entity. While some companies make their codecs widely available, eg. iLBC from Global-IP Sound, other codecs bear a royalty fee.

Another issue that requires attention is the difference between passthrough and transcoding. Passthrough refers to the traversing of media packets through an Asterisk server, without the Asterisk server actually interfering with the media. For example, imagine that Asterisk was connected between two UAs capable of using the G.729A codec. While Asterisk wouldn't be able to understand the media, it would be fully capable of simply connecting the two phones and passing the data as it is. Transcoding refers to the possibility of converting one codec into another. For example, when a call traverses from an IP phone utilizing the G.729A codec to the PSTN, the call is transcoded to the G.711 codec, which is the normal codec of the PSTN network.

Asterisk—The Open-Source PBX

For a successful technology, reality must take precedence over public relations, for Nature cannot be fooled. — Richard P. Feynman

Asterisk, the open-source PBX, was created out of need. Mark Spencer, the creator of Asterisk needed a specialized PBX system to enable the business model of "Linux Support Services", his open-source support services company. This was the truth, back in 1999, when Asterisk was born. Mark required a system to receive voice mail messages and give out a message to the nearest support person. Since he had only \$4000 in his pocket, being a code hacker as he is, he simply sat down and wrote his own system.

According to Mark, Asterisk got its name from the all inclusive UNIX wildcard, the * symbol. Mark imagined Asterisk as an all inclusive telephony platform, which would be able to perform more than just telephony.

In accordance with Richard P. Feynman's saying, Asterisk filled a niche in an area where people had been mostly frustrated with over the course of the years. Telephony platforms were always the domain of the vendors and integrators, with support structures and services that were all profitable to the provider and usually ended with the customer being depressed. Asterisk changed that; suddenly, IT managers capable of installing and managing a Linux server were building their office PBX systems with Asterisk, throwing away traditional PBX systems. Indeed, Nature could no longer be fooled by the vendors and Asterisk had slowly, but surely, started taking its precedence in the world.

Asterisk is Dually Licensed—What Does it Mean?

While the most common variant of Asterisk will use a GPL license, Asterisk is also available under a commercial license. Without going into the discussion of what GPL is, or if Asterisk complies with it, examine the following situations in which the dual license becomes an issue.

According to the GPL license, if one downloads Asterisk, modifies and uses it for one's personal use, one is encouraged to release the modification made to Asterisk, to the community.

On the other hand, if somebody downloads Asterisk, modifies it, repackages and resells it to a customer, they are obliged to release the code either to the customer or the community. Under the same rule, if somebody embeds Asterisk into an existing product they are already marketing, they are obliged to either pay a license fee to Digium, or make their product code available to the community.

A question may arise, "There are various companies marketing Asterisk PBX systems; have they purchased a license? Or are their products GPL-bound product?" – Well, the answer isn't that simple. While Asterisk enjoys a dual license, there exists a possibility of using Asterisk in your product, repackaging it, while still keeping your product code as proprietary code. That is done via utilization of the AGI and AMI interfaces. AGI and AMI enable the creation of Asterisk-enabled logic, while keeping your code base complete separated from Asterisk. This means that a product developed, utilizing AGI and AMI, will be a proprietary product, although the underlying Asterisk is actually a GPL product.

Enter the Asterisk—the Future is Here

Asterisk has introduced a new way of doing business, affecting the telecommunications market, spanning the entire spectrum. While in the past high-priced telephony environments were required in order to create a call center, Asterisk has provided a simple, cost-effective, and most importantly – open solution, in comparison to the other solutions.

Since the year 2004, Asterisk-based services and products have been constantly introduced to the market, starting with AstLinux, Asterisk@Home (aka: TrixBox), through A2Billing, to complete services environments like Rebtel and Jajah – the ever popularity of Asterisk could no longer be ignored by the industry.

Asterisk is currently being deployed in multiple environments, both in the carrier and the enterprise. Where a traditional carrier would have installed a vendor-specific IP Telephony platform or an IP Centrex platform, carriers are building their own infrastructure based upon the Asterisk project, lowering their overall TCO (Total Cost of Ownership) and increasing their manageability.

AsteriskNOW—The Asterisk Software Appliance

Asterisk® in minutes. AsteriskNOW is an open source Software Appliance; a customized Linux distribution that includes Asterisk (the most popular open source IP PBX in software), the AsteriskGUI™, and all other software needed for an Asterisk system. AsteriskNOW is easy to install, and offers flexibility, functionality and features not available in advanced, high-cost proprietary business systems.—<http://www.asterisknow.org>.

Ever since the creation of the Asterisk project, the importance of a proper management GUI has been clear to everybody. Initially, Asterisk had no preset GUI, which opened the work for other people. Various GUI applications such as AMP (Today: FreePBX), VoiceOne, De-Star, and others tried various approaches; however, these approaches were so different, making each of these solutions a fairly closed solution. In addition, due to the nature of the Asterisk project, all these GUIs had to keep up with the Asterisk project, sometimes, they were simply unable to keep pace with Asterisk. For example, while FreePBX works well with branch 1.2, it is not fully compatible with branch 1.4 of Asterisk (at the time of writing this book).

As a result, as part of Digium's vision (and mostly Mark's), Mark started working on a standardized web-based GUI, to be directly interfaced with the Asterisk configuration core. Thus, the creation of AsteriskNOW was upon us. AsteriskNOW provides a simple, easy-to-install, no-fuss solution for getting up and running fast with Asterisk. While several other open-source projects boast of being able to get Asterisk up and running in 20 minutes, it had been proven that it is never the case. This doesn't suggest that there is no need to understand how VoIP and the telephone network work, in order to get things up and running; AsteriskNOW is simply a faster way to get up and running with Asterisk as a VoIP/TDM Hybrid PBX system.

However, instead of talking about AsteriskNOW, this book will serve you as a cook-book and reference book for AsteriskNOW. Our objective is for you to be able to install a fully functional PBX system, using the AsteriskNOW distribution. This book will guide you step by step, from installing the actual distribution, initial configuration, basic extension configuration, routing logic through IVR (Interactive Voice Response—or as referred to by some people: "the automatic attendant") creation up to queue management—you will learn how to build your own PBX.

Summary

At this point you should have got a basic understanding of telephony terms and concepts, and some preliminary information regarding Asterisk—the Open Source PBX. While you are now equipped with information, this information still doesn't enable you to progress on your journey of installing your first AsteriskNOW-based PBX system. Chapters 2 through 10 will guide you through the initial setup of your PBX system. In addition, each chapter will present some advanced options available through AsteriskNOW's web-based administration interface. Chapter 11 will present advanced configuration issues and concepts, while Chapter 12 will present advanced topics in Asterisk in general.

2

Building a PBX

Through the course of this book, you will be guided through the process of installing and performing preliminary configuration on your AsteriskNOW-based PBX. In order to do so, you must outline what your project is, what it will consist of, and most importantly what is your final objective.



While this book serves as a practical guide to AsteriskNOW, it must not be mistaken for an Asterisk crash course. Learn more about Asterisk via certified Asterisk courses offered by Digium or Digium authorized training partners in your country.

Objective—Building an Office PBX

Start with a clear description of your project's objective. Your objective is to build an office PBX utilizing the AsteriskNOW software appliance. In order to do so, the various aspects of your future installed system are described in a form that would usually be used by professional PBX installers.

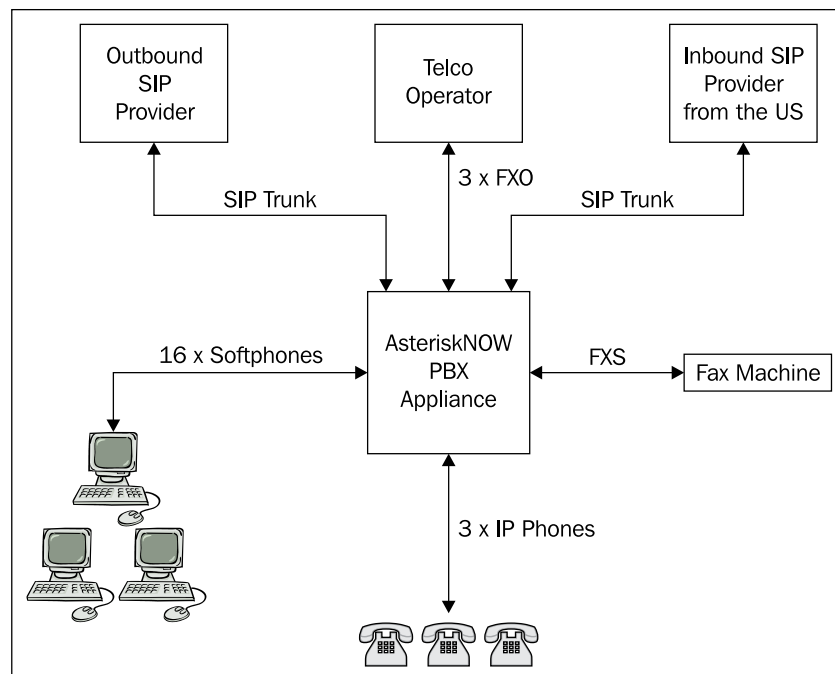
Customer Name	Fictitious Creations Ltd.
Customer Hardware	Intel Pentium 4 Desktop PC 512 MB RAM 80 GB Hard Drive (please note that the hard drive is either a blank hard drive, or one you can spare – as it's going to be deleted) Digium TDM400 with 3xFXO modules and 1xFXS module
Number of trunks	3 x analog trunks 1 x SIP based trunk for international calls 1 x SIP based trunk for inbound DDI from the US
Number of analog extensions	1 x Internal analog extension, used for fax machine

Customer Name	Fictitious Creations Ltd.
Number of IP extensions	16 x IP Soft phones 2 x LinkSys 941 IP Phone 1 x SNOM 320 IP Phone
Inbound Routing	Call intercepted by any of the trunks should be directed into the main IVR menu.
Outbound Routing	Any cellular number (07XXXXXXXXX) should be directed to FXO module 1. All other local numbers should be directed to FXO module 2. All international calls should be routed to the international SIP trunk. All faxes should be directed to FXO module 3.

Physical Connectivity

Like any other networking-enabled environment, it is imperative that a schematic of all connections to and from the PBX system is drawn. This stage is essential, as it will serve as a blueprint for defining the various inbound and outbound routes later on.

To make it simple, use a block diagram to describe your system; later, other information could be added to it.



As you can see, the diagram includes all the inbound and outbound routes available to your PBX in addition to the various internal IP and FXS connections. For the time being, devise a table indicating each device on your system, so that you will have a clear view of the installation scenario.

PBX Information

- IP Address: 82.14.29.13
- Netmask: 255.255.255.0
- Default GW: 82.14.29.254

Please note that while the above indicates a public IP address, the rest of the book indicates that the PBX is located on a private IP address. In general, for your configuration to work, you would have to configure a static NAT from your public IP address to your PBX private IP address.

Trunk Information

Telco Operator Information:

- 3 connections of FXO interface (channel 1, channel 2, channel 3)

SIP trunk Configurations:

- Outbound SIP Provider
 - IP Address: 62.116.43.1
 - Signalling: SIP
 - Codecs: ulaw, alaw, gsm, ilbc
- Inbound DID Provider
 - IP Address: 212.134.1.3
 - Signalling: IAX2
 - Codecs: ulaw, alaw, gsm

Internal Connectivity

- Physical Phones
 - 1 x FXS connection for a fax machine (channel 4)
 - IP Phones – Soft Phones
 - 16 x IP Soft phones (SIP based IP soft phones, using CounterPath EyeBeam)
 - IP Phones – Hardware Phones
 - 2 x IP Receptionist phone (SIP based IP hard phone, using SNOM 320)
 - 1 x IP Guest phone (SIP based IP hard phone, using LinkSys 941)
-

Installation Procedure Outline

The installation procedure includes five specific stages – downloading, burning, booting, preliminary configuration, and rebooting.

Downloading AsteriskNOW



The AsteriskNOW distribution is available from the AsteriskNOW website, located at <http://www.asterisknow.org>. Please note that at the time of writing this book, AsteriskNOW was at version Beta 6.

AsteriskNOW images are available in several formats and image types:

AsteriskNOW (32-bit)

The 32-bit image download is intended to be used on hardware platforms compatible with the 32-bit processor and motherboard paradigm. This image can also be installed on 64-bit capable motherboards and processors; however, it will be less optimized. This image works well with Pentium-3 and Pentium-4 based computer.

AsteriskNOW (64-bit)

The 64-bit image download is intended to be used on hardware platforms compatible with the 64-bit processor and motherboard paradigm. This installation image will work only with 64-bit capable motherboards and processors. This image works well with newer model Pentium-4 and Pentium-4 Dual Core.

AsteriskNOW (x86 xen image)

The XEN x86 image is intended to operate on a server using a Xen Universal guest domain. To learn more please visit the XEN website (<http://www.xensource.com>).

AsteriskNOW (x86 VMware image)

VMware is a well-established Virtual Machine environment for Windows and Linux. The VMware image is a pre-installed AsteriskNOW appliance, which can be directly loaded into a VMware server, workstation or player and to start working with it. For more information about VMware, please visit the VMware website (<http://www.vmware.com>).

AsteriskNOW (x86 LiveCD)

The LiveCD lets you experience AsteriskNOW without actually installing anything on your computer. Once the LiveCD is booted, a fully working image of AsteriskNOW is operational on your computer, which you can configure and work with.

AsteriskNOW Hardware Requirements

While AsteriskNOW will work on most modern PCs (Pentium III-based PC is suggested), you must take time to know AsteriskNOW-specific hardware, mainly Digium-based hardware, which is automatically supported by AsteriskNOW.



Various Asterisk-compatible hardware exists in the open market, mainly from companies like Sangoma, Yeaster, and ATCOM. While this hardware can be installed with the downloadable version of Asterisk (which requires patching of the Zaptel kernel module), installation with the AsteriskNOW distribution isn't straightforward mainly due to the fact that the AsteriskNOW distribution doesn't include the AsteriskNOW sources.

Discussion on installing alternative hardware is beyond the scope of this book—information may be made available, by the various vendors, in the future.

Analog Interface Cards

Digium provides analog interface cards, which come in several models—each one with its own sizing. Sizes vary from 4 port cards through 8 port cards up to 24 ports on a single card.

The following table describes each of the available interface cards, with its main characteristics:

Model	Description	Available Modules	Configurations
TDM400P	A four-port analog interface card, with exchangeable modules.	X100 – FXO	4 x FXO, 0 x FXS
		S100/110 – FXS	3 x FXO, 1 x FXS
			2 x FXO, 2 x FXS
			1 x FXO, 3 x FXS
			0 x FXO, 4 x FXS
TDM800P	An 8-port analog interface card, with exchangeable modules.	X400 – 4 x FXO	4 x FXO, 4 x FXS
		S400 – 4 x FXS	4 x FXO, 0 x FXS
			0 x FXO, 4 x FXS
			8 x FXO, 0 x FXS
			0 x FXO, 8 x FXS
TDM2400P	A 24-port analog interface card, with exchangeable modules.	X400 – 4 x FXO	12 x FXO, 12 x FXS
		S400 – 4 x FXS	8 x FXO, 16 x FXS
			4 x FXO, 20 x FXS
		Echo Cancellation Module	0 x FXO, 24 x FXS
			16 x FXO, 8 FXS
			20 x FXO, 4 FXS
			24 x FXO, 0 x FXS

Digital Interface Cards

Digium provides digital interface cards, which come in several models – each one with its own size. Sizes vary from 4 Basic Rate Interfaces (BRI) through a single Primary Rate interface (PRI) up to 4 PRI interfaces.

The following table describes each of the available interface cards, with its main characteristics:

Model	Description	Voltage
TE110P	Single-Span E1/T1 PRI interface	110P – PCI 5V/3.3V
TE120P		120P – PCI 5V/3.3V
TE122		122 – PCI 5V/3.3V
TE121		121 – PCI-E
TE210P/TE212P	Dual-Span E1/T1 PRI interfaces 210P and 205P do not include hardware echo cancellation 212P and 207P include hardware echo cancellation	TE210P – PCI 3.3V
TE205P/TE207P		TE212P – PCI 3.3V
TE220		TE205P – PCI 5V
TE420		TE207P – PCI 5V
		TE220 – PCI-E
		TE420 – PCI-E
TE410P/TE412P	Quad-Span E1/T1 PRI interfaces 410P and 405P do not include hardware echo cancellation 412P and 407P include hardware echo cancellation	TE410P – 3.3V
TE405P/TE407P		TE412P – 3.3V
		TE405P – 5V
		TE407P – 5V
B410P	Quad-Span BRI interfaces with hardware echo cancellation	5V/3.3V

Additional Add-On Cards

Recently, Digium added a new transcoder card, named TC400B. The TC400B card is described as a dedicated DSP (Digital Signal Processing) resource, capable of off-loading the Asterisk-based codec operations to hardware, thus, lowering the general utilization of the CPU running an Asterisk-based VoIP system.



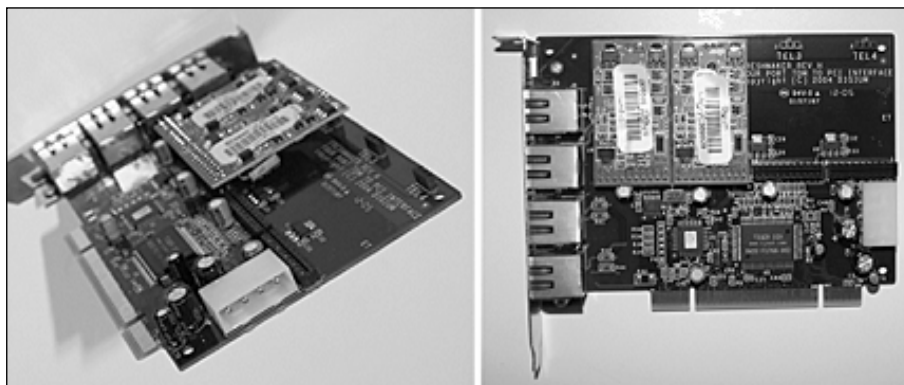
In this book, a TDM400P card equipped with three X100 modules and a single S110 module is used.

The Installation Process

If you've ever installed a Linux server or workstation, the AsteriskNOW installation process is no different.

Step 1: Hardware Installation

Install the newly purchased Digium hardware. If you're using a TDM400P based card, your newly purchased hardware would look like the following figure:

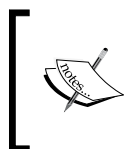


Please note the power inlet located at the lower right side of the interface card. This power inlet must be connected to a power cable inside your computer, in order to allow the FXS interface to operate.

If your installation doesn't require an FXS interface, you may leave this inlet unconnected. The card must be installed into a PCI-compliant (either 3.3V or 5V) slot in your computer. If you are unsure about the installation of this card, allow a qualified technician to install it for you.

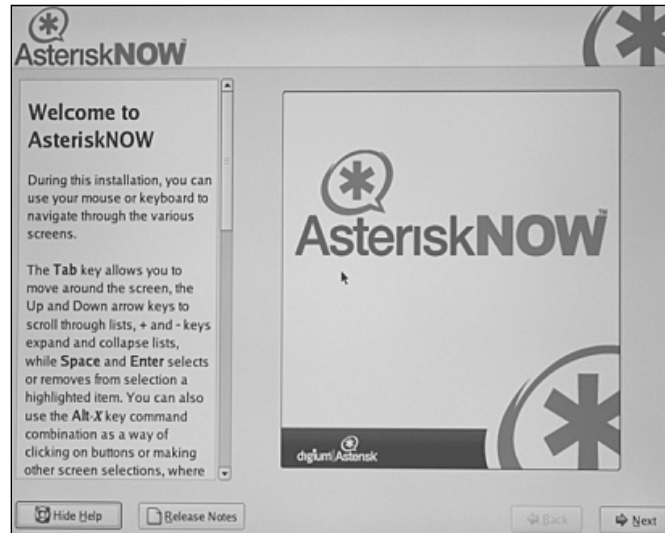
Step 2: Install the AsteriskNOW Distribution

At this point, your computer should already be assembled with the Digium interface card, and you should be ready to install the AsteriskNOW distribution.



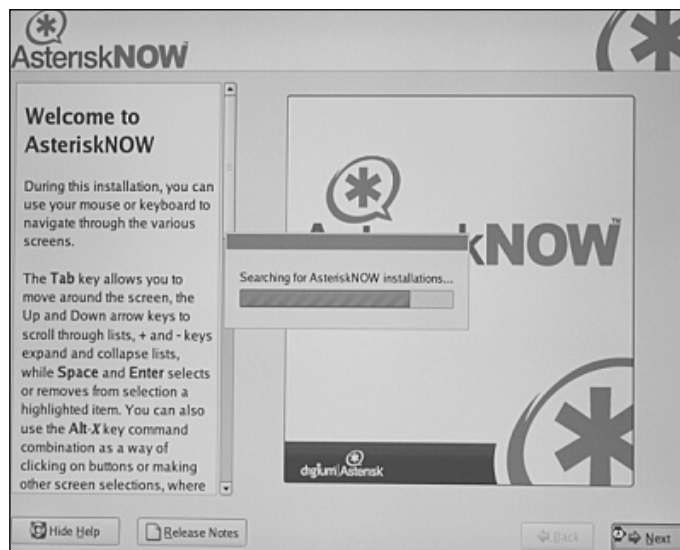
AsteriskNOW is based upon a distribution called rPath (<http://www.rpath.com>). The rPath distribution closely resembles the RedHat/Fedora installation process—if you are familiar with it, you may skip to the next section.

Once you have booted your AsteriskNOW distribution, the following screen should be observed:



From this point, the installation is fairly automatic and you have to follow the installation wizard. The installation itself doesn't require specific configurations to be made, apart from the network configuration and login information, which will be explained shortly.

Once you click the **Next** button, the following screen should appear:

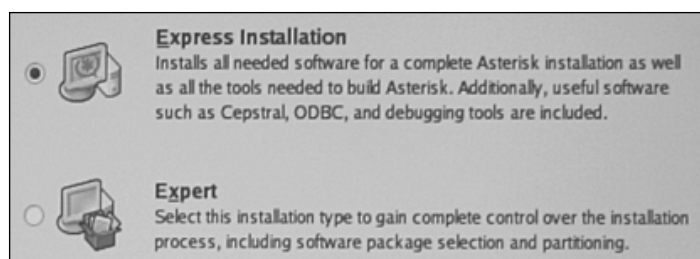


AsteriskNOW will now search for the available AsteriskNOW installation image.



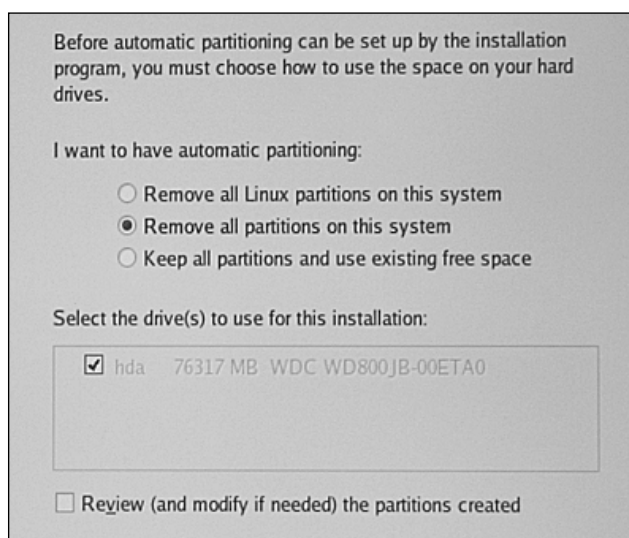
At the time of this writing, the current version of AsteriskNOW is Beta 6. By the time this book is published, the release version of AsteriskNOW could be available to the public.

Once the installation system has found the proper AsteriskNOW image to install, you will be presented with the option of performing an **Express Installation** or an **Expert Installation**.



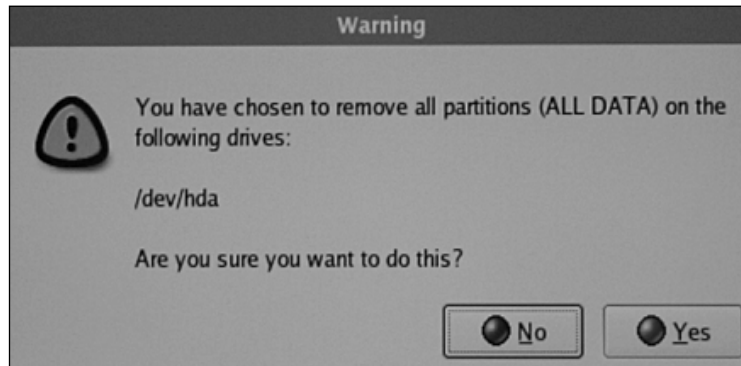
The main advantage of an **Expert** installation is that it allows complete control over the installation process, including software package selection and partitioning. For most installations, the Express method of installation would be preferred.

At this point, select the **Express Installation** option, and click the **Next** button, the following should now appear on your screen:



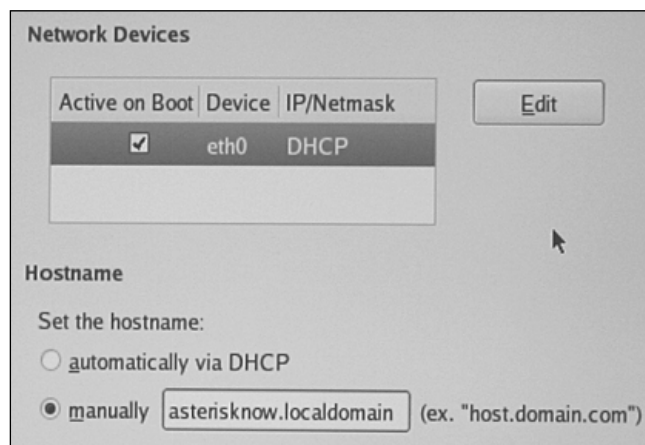
AsteriskNOW will now partition your hard drive for its needs.

For the purpose of your installation, assume that the hard drive can be fully deleted, and you will allow AsteriskNOW to configure everything for you. Select the **Remove all partitions on this system** option, and click the **Next** button. The following screen should be observed:



Word of caution: From this point your hard drive will be deleted; if it contained information, you won't be able to restore it—be careful!

Click the **Yes** button to indicate that you would like to proceed with the installation. The following should now appear on your screen:



Please note that in a normal office environment, it is not recommended to install the PBX using a DHCP network mode. The reason is that you would require your VoIP terminals and trunks, be they SIP or IAX2, to connect to a well-defined IP address, thus, a static address will be required. If you are unsure as to what IP address to use, consult with your network administrator for the proper IP address information.



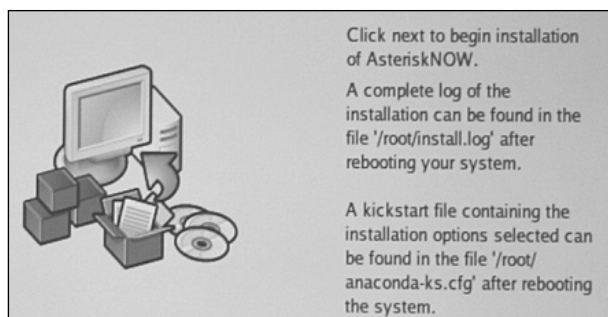
For the remainder of this book, assume that the network configuration is as follows:

Network Address: 192.168.2.0
Network Mask: 255.255.255.0
Default Gateway: 192.168.2.254
PBX IP Address: 192.168.2.1

Once you have established your network configuration, click the **Next** button. The following screen should be observed:

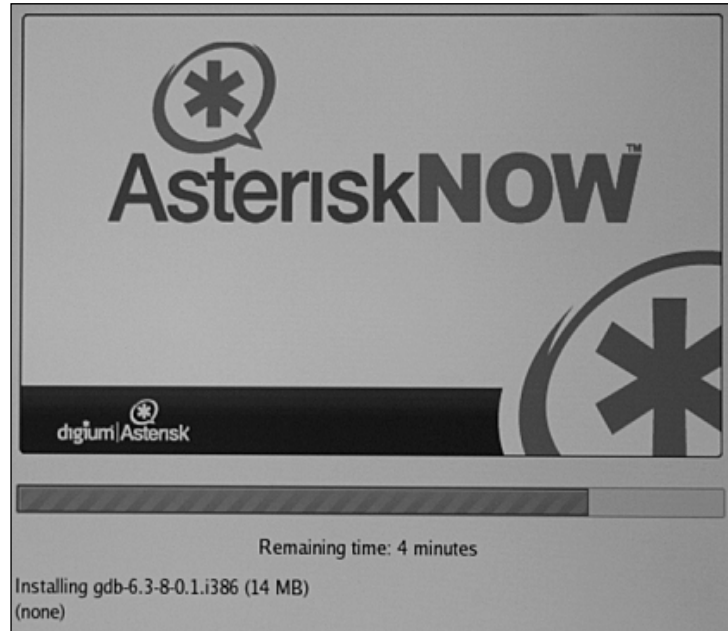
If you are familiar with the process of installing a Linux distribution, the above screen may be misleading — **This is not the root password!**

While the dialog box is identical, the password being set in this dialog is for a user named **admin**. The admin user is used through the web GUI of AsteriskNOW and the administration of the rPath appliance. When setting up this password, make sure that you write this password down somewhere. Once you have set the admin password, the following screen should be observed:



Once you click on the **Next** button, the distribution will start installing the AsteriskNOW distribution. At this point, you can sit back and relax; the rest is automatic. Depending on your computer, the installation may take anything from 12 to 35 minutes.

The following screen should be observed:



Once the installation is complete, the following screen should be observed:



Click on the **Finish** button shown at the lower right sight of your screen. Once your computer reboots, the following screen should appear, indicating the AsteriskNOW boot loader:

```

GNU GRUB version 0.95 (639K lower / 521408K upper memory)

AsteriskNOW (2.6.19.7-0.1.smp.x86.i686.cmov)

Use the ↑ and ↓ keys to select which entry is highlighted.
Press enter to boot the selected OS, 'e' to edit the
commands before booting, 'a' to modify the kernel arguments
before booting, or 'c' for a command-line.

The highlighted entry will be booted automatically in 1 seconds.

```



If you were to examine the above boot loader screen, you would surely notice that the kernel version appears in the menu. The currently installed kernel on the demo server is version 2.6.19, with SMP support and compiled for the i686 platform. While this is true for the demo server, the indication on your server may differ, depending on the downloaded distribution and the hardware used for the installation.

Once you have waited the traditional 5 seconds (or pressed on the enter key for the impatient ones), your screen will start showing a normal Linux startup. If you are familiar with the Linux operating system, the following section won't look strange to you, apart from the following section:

```

Starting kernel logger: [ OK ]
Starting NFS statd: [ OK ]
Loading zaptel framework: [ OK ]
Waiting for zap to come online...OK
Loading zaptel hardware modules: tor2.
wct4xxp.
wct1xxp.
wctel1xp.
wctdm24xxp.
wcfxo.
wctdm.
wcusb.

Running zapscan: Scanning for zaptel devices...OK
Running ztcfg: [ OK ]
Mounting other filesystems: [ OK ]
Starting sshd: [ OK ]

```

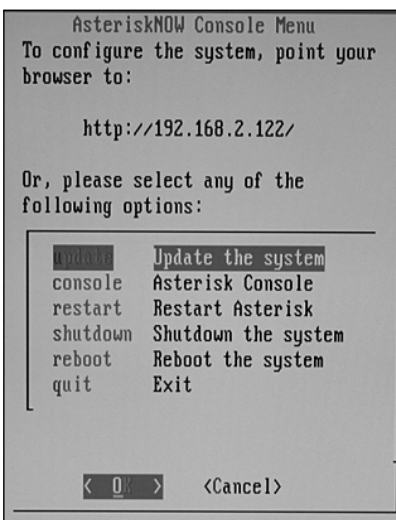
The highlighted area of the start-up sequence indicates the loading of Digium hardware modules. Traditionally speaking, AsteriskNOW will support any Digium-compatible hardware; that is to say, if you intend to use a different type of hardware, you may surely run into issues.



You could use hardware of your choice, but remember that the AsteriskNOW distribution is backed and developed by Digium, which means that it will most probably work the best with Digium-based hardware, or hardware fully compatible with Digium hardware.

Hardware products that may require additional patches to the Zapata kernel module will most probably require some out-of-band compilation and modifications.

Once your start-up sequence is fully completed, the following screen should appear on your screen, indicating that the AsteriskNOW appliance is now ready for usage:



This screen indicates that in order to configure the Asterisk PBX or the rPath distribution, you need to point your browser to the IP indicated. In the above example the PBX was configured to utilize the IP number — **192.168.2.122** — however, your installation may vary according to the IP you provided during the installation phase (or one you have been automatically assigned by your network DHCP server).

While the above screen is fairly simple, one useful option would be the **Asterisk Console** option. The Asterisk console displays the workings of your PBX. While calls traverse the PBX, your console screen will be constantly updated with various status reports and debug information.

The following screenshot shows what the Asterisk Console screen may look like:

```

Registered custom function ENV
Registered custom function STAT
(func_env.so => (Environment/filesystem dialplan functions))
Registered application 'Dictate'
app_dictate.so => (Virtual Dictation Machine)
Parsing '/etc/asterisk/skinny.conf': Found
Skinny listening on 0.0.0.0:2000
Registered channel type 'Skinny' (Skinny Client Control Protocol (Skinny))
Registered application 'Transfer'
app_transfer.so => (Transfer)
Registered application 'SendURL'
app_url.so => (Send URL Applications)
Registered application 'SoftHangup'
app_softhangup.so => (Hangs up the requested channel)
Registered custom function MATH
(func_math.so => (Mathematical dialplan function))
Registered application 'Pickup'
app_directed_pickup.so => (Directed Call Pickup Application)
Registered application 'SetTransferCapability'
app_settransfercapability.so => (Set ISDN Transfer Capability)
Registered application 'TestClient'
Registered application 'TestServer'
app_test.so => (Interface Test Application)
Asterisk Ready.
*CLI> _

```

You have successfully completed the first section of the installation process. Now you need to perform the initial appliance configuration, which is explained in the following section.

Step 3: The Initial Configuration

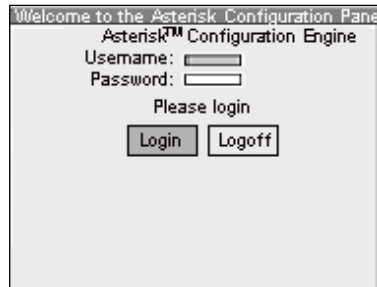
Initial configuration is performed using a regular web browser, from any PC (Windows or Linux based).



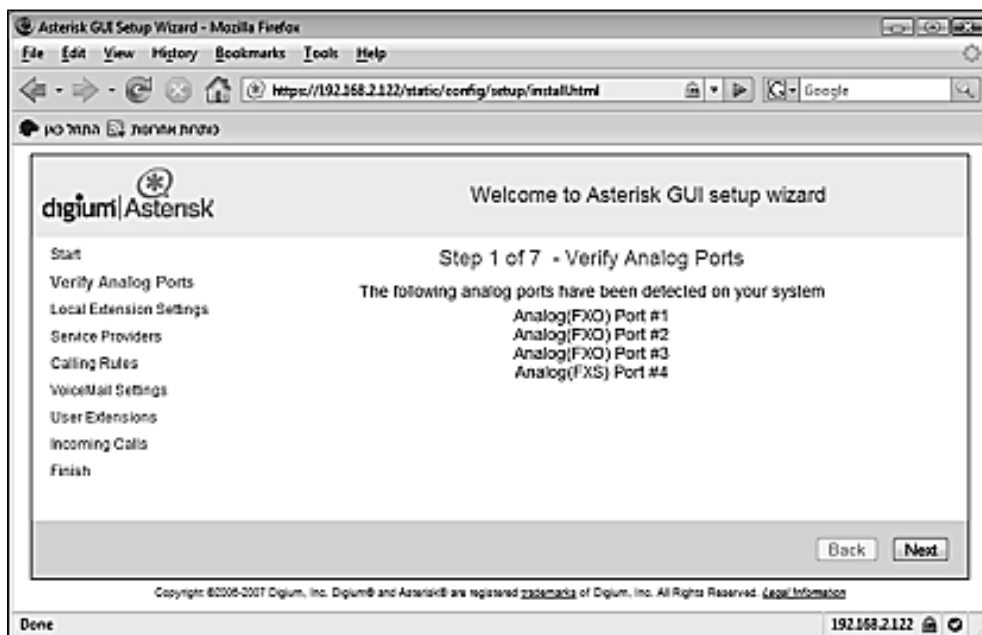
Important Note: The AsteriskNOW GUI currently operates with the Mozilla Firefox web browser – not with Microsoft Internet Explorer. The reason for this is located in the way these browsers implement the JavaScript language, which is used extensively with the AsteriskNOW GUI. You can download the Mozilla Firefox browser at <http://www.mozilla.org>.

Once you start your web browser and point it to your newly installed server, you are sure to receive an error indicating that your browser has received an invalid certificate; this is normal. The AsteriskNOW configuration GUI is encrypted using a self-signed encryption key, thus, you receive this message.

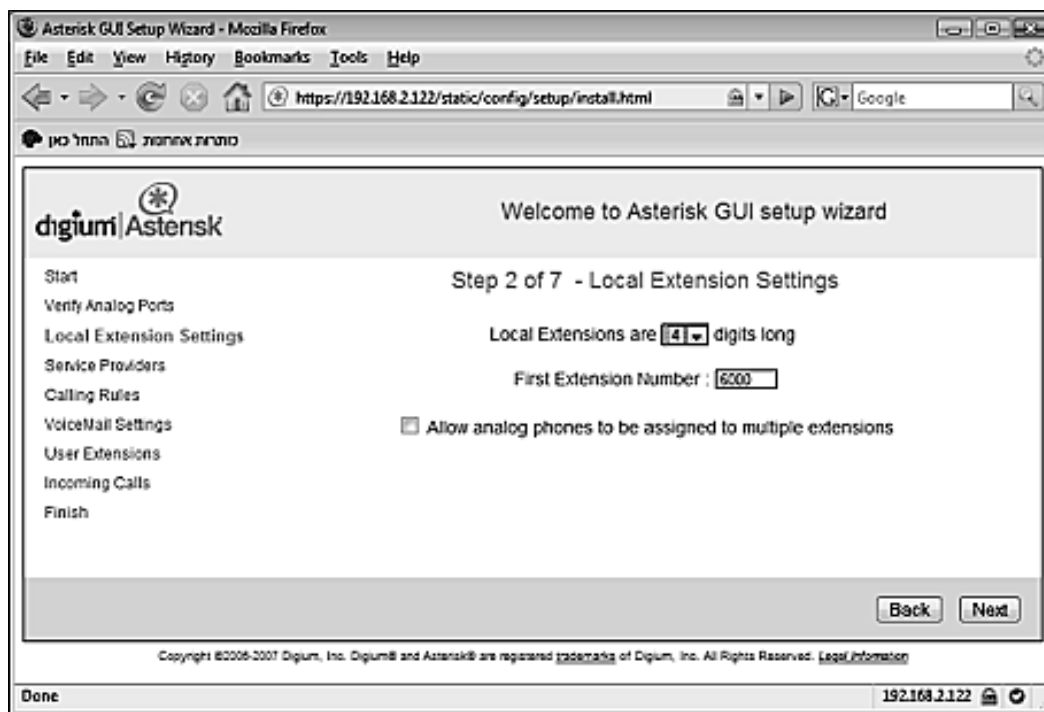
Simply accept the certificate received and press the **OK** button; your browser will now greet you with the following screen:



You are requested to provide the admin username and password you configured during the installation process. Simply enter them in the respective fields and click the **Login** button. The following screen should now appear:



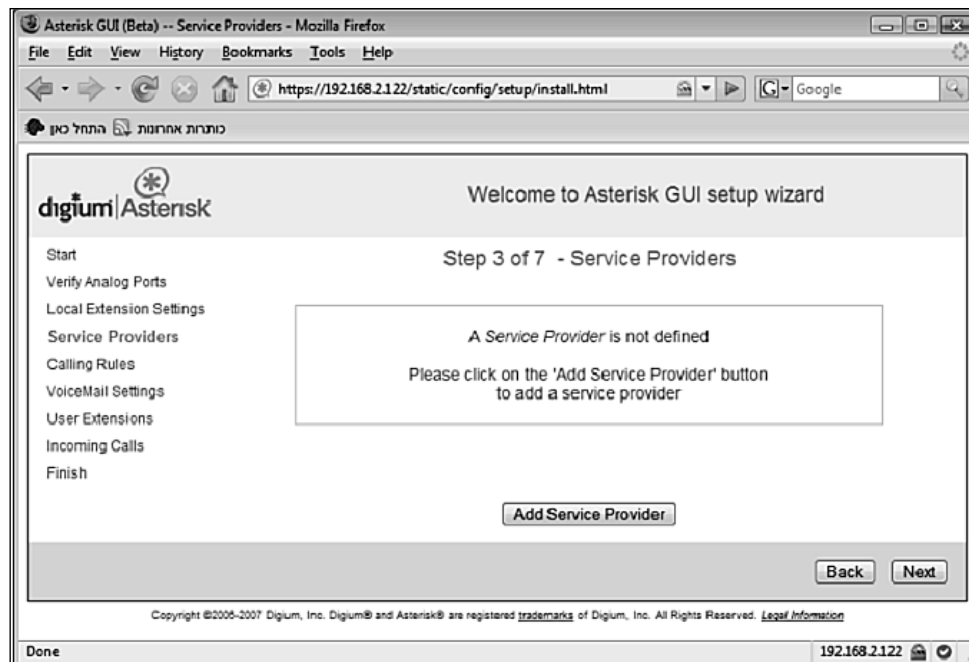
Depending on your installation, your screen will now show the number of ports that were identified during the system initial start-up and configuration. Your system consists of three FXO modules (Ports 1, 2, and 3) and a single FXS module (port 4). Verify that your detected configuration matches the hardware you've installed, and click the **Next** button at the bottom right section of the dialog box; the following screen should be observed:



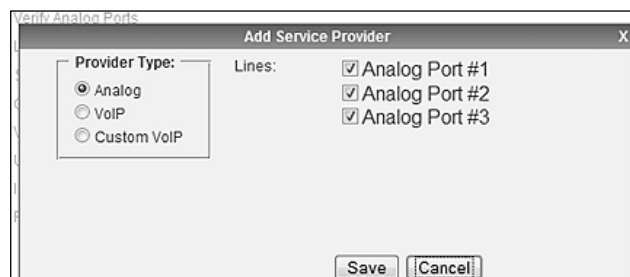
The above screen indicates your local extensions numbering space. The number of digits are defined, as the length of your extensions number will determine the maximum number of extensions available on your PBX. For example, if defined as 4 digit extensions, the first extension of your PBX may be 1000 and the last may be 9999 – indicating that your PBX can configure up to 8999 extensions. As the AsteriskNOW GUI automatically advances the extension number every time you create a new one, the initial configuration asks you what should be the first extension on your PBX. As an example indicate that your PBX will utilize four-digit extension lengths, the first extension will be 6000, and that you don't allow your analog phones to be assigned to more than a single extension.

[ While it is possible to configure AsteriskNOW analog phones (FXS ports) to be associated with several logic extensions, this functionality will not be utilized during the course of this book.]

Once you have defined your **Local Extension Settings**, click the **Next** button to continue to the **Service Providers** section; the following screen should be observed:



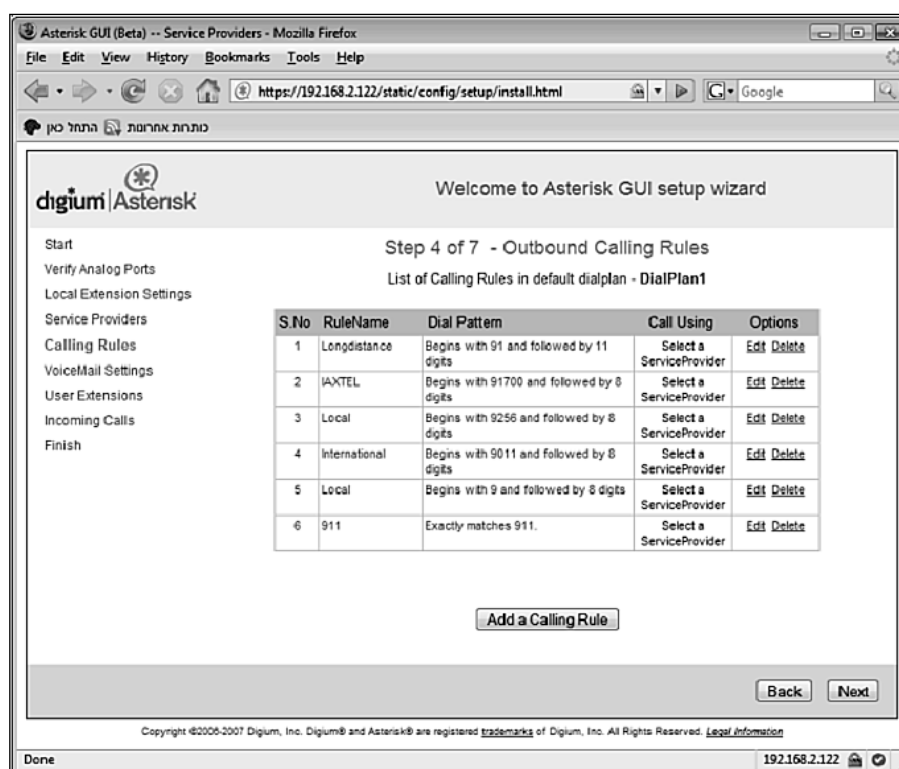
At this point, you need to define your service providers. In accordance with the above information, you need to establish a connection to a service provider. For the initial configuration, create a single service provider, indicating the three analog connections of our server (3 x FXO interfaces). Click the **Add Service Provider** button, and select an **analog** provider to be added. The following screen should be observed:



Make sure you select all three analog ports to be a part of this analog service provider. Once you have clicked the **Save** button, your service provider screen should show the following:

S.No	Service Provider	Type	
1	Ports 1,2,3	Analog	Edit Delete

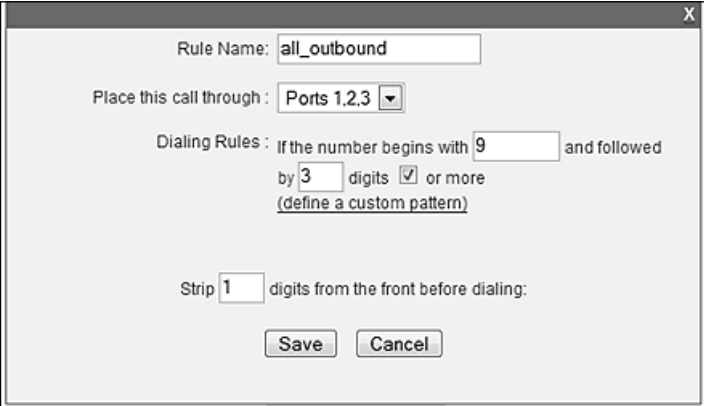
You have created service provider ID 1, which consists of analog **Ports 1, 2 and 3**. Click on the **Next** button to proceed. The following screen should be observed:



The above screen indicates the various pre-configured routes available with our AsteriskNOW distribution. As you can see, all routes are indicated by a message – **Select a Service Provider**. For the time being, perform the following actions:

- Delete all the available routing rules.
- Create a single routing rule, consisting of any number that begins with the digit 0, with any number of digits following, to be routed to the analog service provider.

The following is a screenshot of the configuration screen and the final result:



Rule Name:

Place this call through:

Dialing Rules : If the number begins with and followed by digits ☒ or more
(define a custom pattern)

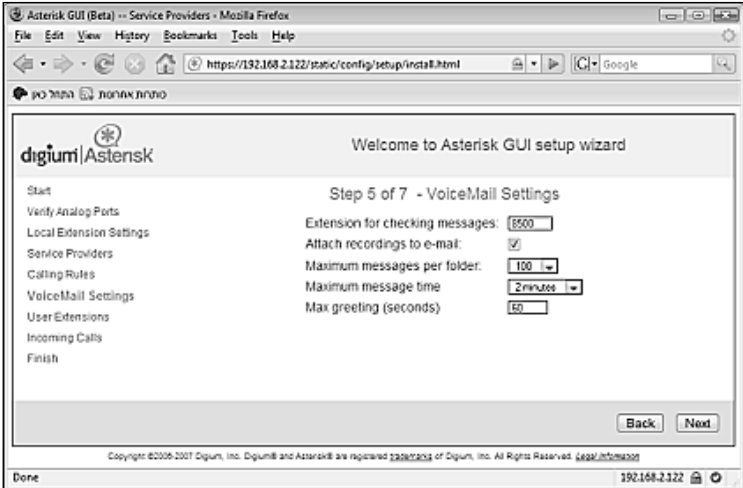
Strip digits from the front before dialing:

↓

S.No	RuleName	Dial Pattern	Call Using	Options
1	all_outbound	Begins with 9 and followed by 3 or more digits	Ports 1,2,3	Edit Delete

According to your configuration, the PBX would route any number of the form 9+(3 digits or more) to the analog service provider. This means that any number of the form 9+(2 digits or less) will not be served by our PBX.

Click on the **Next** button to continue our initial configuration. The following screen should be observed:



Asterisk GUI (Beta) - Service Providers - Mozilla Firefox

File Edit View History Bookmarks Tools Help

https://192.168.2.122/static/config/setup/install.html

Welcome to Asterisk GUI setup wizard

Step 5 of 7 - VoiceMail Settings

Start

Verify Analog Ports

Local Extension Settings

Service Providers

Calling Rules

VoiceMail Settings

User Extensions

Incoming Calls

Finish

Extension for checking messages:

Attach recordings to e-mail: ☒

Maximum messages per folder:

Maximum message time:

Max greeting (seconds):

Copyright ©2006-2007 Digium, Inc. Digium® and Asterisk® are registered trademarks of Digium, Inc. All Rights Reserved. [Legal Information](#)

Done 192.168.2.122

This screen configures your voicemail system general settings. Your main concern at this point of time would be to configure an extension number to access your voicemail system, the maximum length of your messages, the maximum number of messages per voicemail folder, the maximum time for a voicemail greeting, and whether you would like to send your received voicemail as an email attachment to your extension's mailbox. Please configure according to the following information:

Extension for checking messages:	8500
Attach recordings to e-mail:	☒
Maximum messages per folder:	100
Maximum message time	2 minutes
Max greeting (seconds)	60

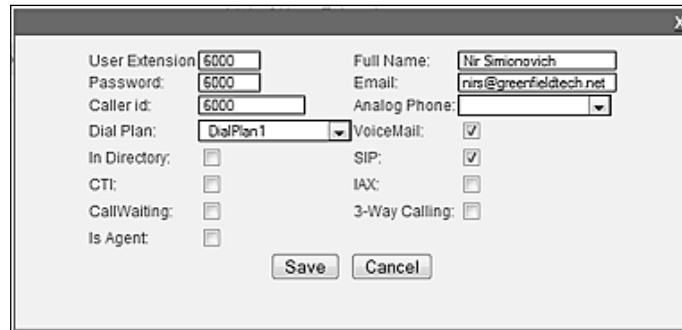
Click on the **Next** button to proceed to the next step. At this point, you need to configure your user extensions. User extensions can either be analog phones directly connected to your PBX system (via FXS ports), or various VoIP-based extensions. At this point, configure your fax extension, which is connected directly to your PBX using an FXS interface. Click the **Add User Extension** button, and complete the dialog box in accordance with the following information:

User Extension:	8700	Full Name:	Fax
Password:		Email:	
Caller Id:	8700	Analog Phone:	Analog Port #4
Dial Plan:	DialPlan1	VoiceMail:	☐
In Directory:	☐	SIP:	☐
CTI:	☐	IAX:	☐
CallWaiting:	☐	3-Way Calling:	☐
Is Agent:	☐		
Save Cancel			



You must be wondering at this point where "DialPlan1" came from. Once you have established your outbound dialling rules, the DialPlan1 dial plan is automatically created for you.

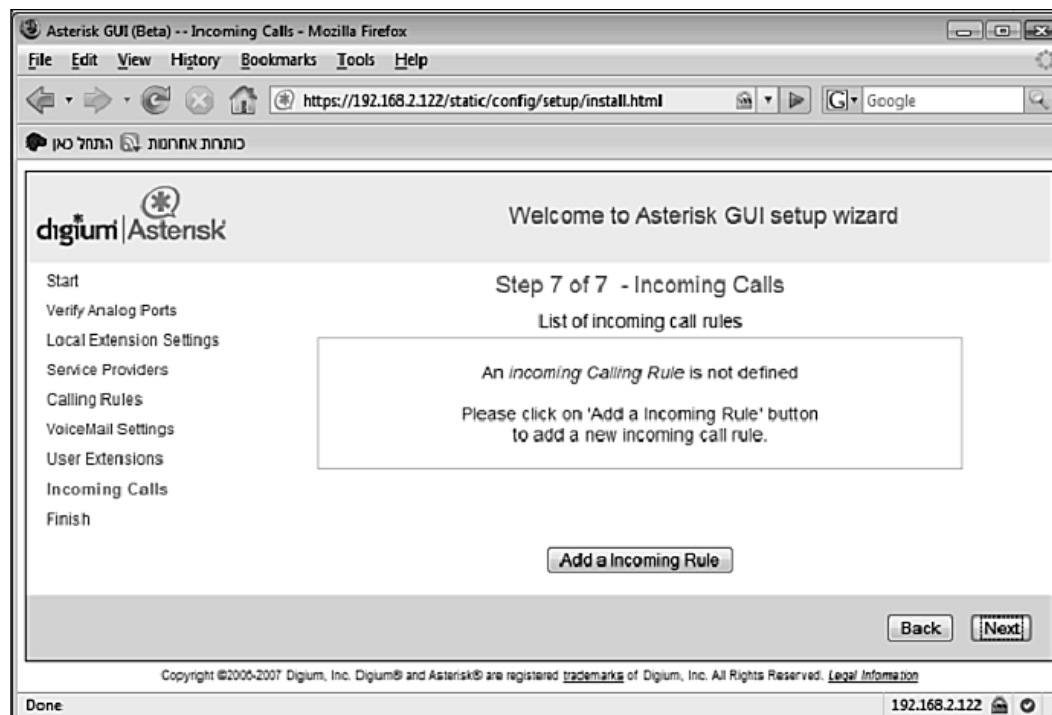
Click the **Save** button in order to save your newly created extension. Now, create another extension, using the following information:



A screenshot of a web-based form for creating a new extension in Asterisk. The form is titled 'X' in the top right corner. It contains two columns of fields. The left column includes: 'User Extension' (text box with '5000'), 'Password' (text box with '5000'), 'Caller Id' (text box with '5000'), 'Dial Plan' (dropdown menu with 'DialPlan1'), 'In Directory' (checkbox), 'CTI' (checkbox), 'CallWaiting' (checkbox), and 'Is Agent' (checkbox). The right column includes: 'Full Name' (text box with 'Nir Simonovich'), 'Email' (text box with 'nirs@greenfieldtech.net'), 'Analog Phone' (dropdown menu), 'VoiceMail' (checkbox checked), 'SIP' (checkbox checked), 'IAX' (checkbox), and '3-Way Calling' (checkbox). At the bottom of the form are two buttons: 'Save' and 'Cancel'.

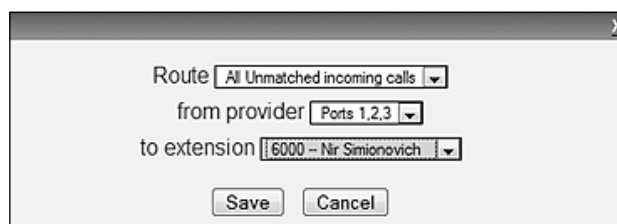
As you can see, this extension is different from the previous one, in that this extension utilizes a SIP flag and also has a voicemail setting. This portion will be explained later.

Click the **Save** button in order to save your newly created extension, followed by a click on the **Next** button in order to proceed. The following screen should appear indicating that you need to configure your inbound routes.



A screenshot of the Asterisk GUI setup wizard in a Mozilla Firefox browser window. The browser title is 'Asterisk GUI (Beta) -- Incoming Calls - Mozilla Firefox'. The address bar shows 'https://192.168.2.122/static/config/setup/install.html'. The page has a sidebar on the left with a list of steps: 'Start', 'Verify Analog Ports', 'Local Extension Settings', 'Service Providers', 'Calling Rules', 'VoiceMail Settings', 'User Extensions', 'Incoming Calls' (which is highlighted), and 'Finish'. The main content area is titled 'Welcome to Asterisk GUI setup wizard' and 'Step 7 of 7 - Incoming Calls'. Below this, it says 'List of incoming call rules' and 'An incoming Calling Rule is not defined'. A message box states: 'Please click on 'Add a Incoming Rule' button to add a new incoming call rule.' Below this message is a button labeled 'Add a Incoming Rule'. At the bottom right of the main content area are two buttons: 'Back' and 'Next'. The footer of the page contains copyright information: 'Copyright ©2005-2007 Digium, Inc. Digium® and Asterisk® are registered trademarks of Digium, Inc. All Rights Reserved. Legal Information'. The browser's status bar at the bottom shows 'Done' and the IP address '192.168.2.122'.

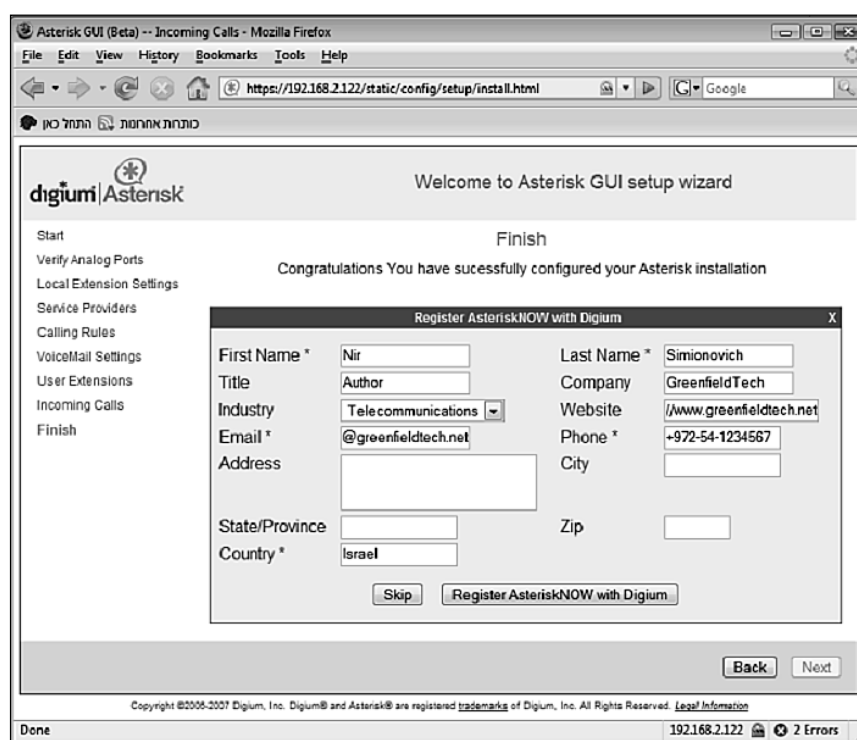
Inbound routes control how calls from your service providers are handled by your PBX. While some calls may require direct routing, other calls may require additional processing. This screen enables you to create a preliminary inbound call-routing logic. Click the **Add a Incoming Rule** buttons. Fill the dialog box with the following information:



Route: All Unmatched incoming calls
 from provider: Ports 1,2,3
 to extension: 6000 - Nir Simionovich
 Save Cancel

According to the above screenshot, any unmatched call (which means any call) coming from the analog service provider will be automatically routed to extension **6000**. At this point, this will allow you to perform most of the preliminary testing you require, so leave it as it is. In the following chapters, you will add interactive voice response (IVR) capabilities and change this behavior.

Click the **Next** button in order to proceed; the following screen should be observed:



Asterisk GUI (Beta) -- Incoming Calls - Mozilla Firefox
 https://192.168.2.122/static/config/setup/install.html

Welcome to Asterisk GUI setup wizard

Finish
 Congratulations You have successfully configured your Asterisk installation

Register AsteriskNOW with Digium

First Name * Nir
 Last Name * Simionovich
 Title Author
 Company GreenfieldTech
 Industry Telecommunications
 Website /www.greenfieldtech.net
 Email * @greenfieldtech.net
 Phone * +972-54-1234567
 Address
 City
 State/Province
 Zip
 Country * Israel

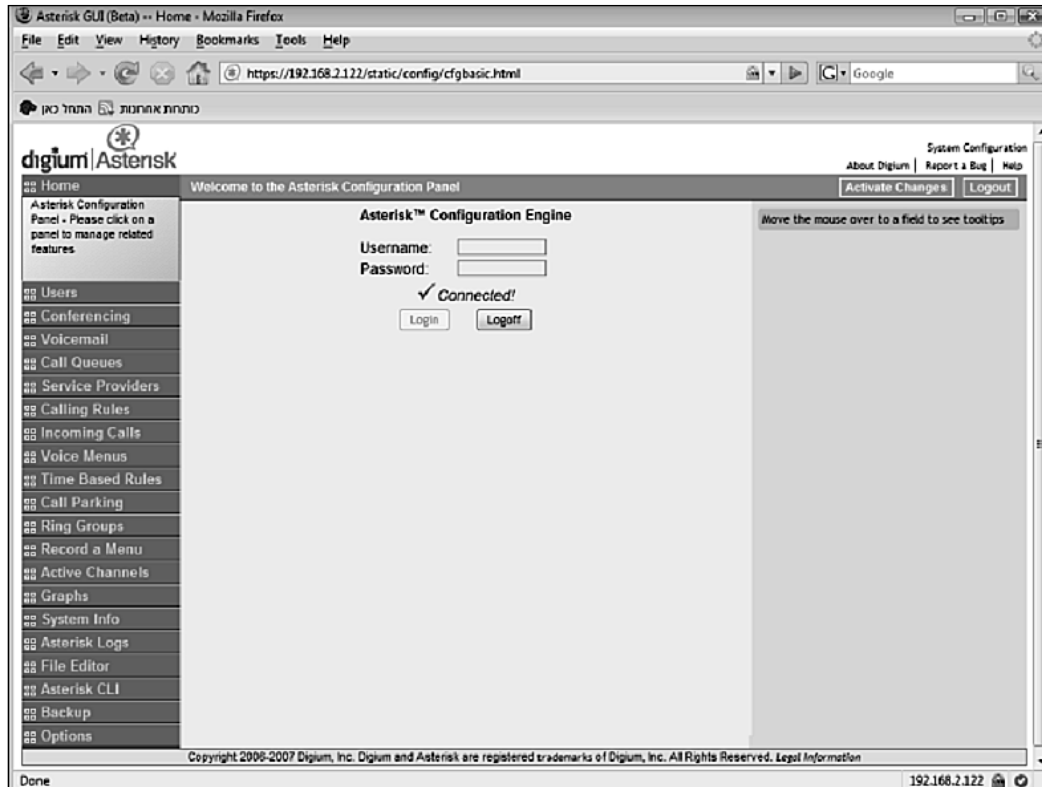
Skip Register AsteriskNOW with Digium

Back Next

Copyright ©2006-2007 Digium, Inc. Digium® and Asterisk® are registered trademarks of Digium, Inc. All Rights Reserved. Legal Information

Done 192.168.2.122 2 Errors

At this point you are basically done, and you may register your copy of AsteriskNOW on the Digium website – as it will assist in promoting the AsteriskNOW project development. For the time being, click the **Skip** button, as this is not your production environment yet.



Congratulations! You have successfully completed the initial configuration of your AsteriskNOW PBX system. At this point, your AsteriskNOW appliance should already be fully working. In the next chapters you will go through the various steps required to configure phones, IP devices, service providers, and the various aspects of your PBX system.

Anatomy of the AsteriskNOW Configuration GUI

As you can surely see from the above image, the AsteriskNOW GUI is separated into two distinct work areas. The configuration menu is on the left, while the workspace is on the right side. Every section on the left is in charge of controlling a different aspect of the AsteriskNOW PBX appliance. Over the course of this book, you will explore each of these sections – learn the various options available to you as administrator and thus create a fully functional PBX system.

Introduction to the rPath Appliance GUI

As mentioned before, the AsteriskNOW distribution is based on the rPath distribution appliance (<http://www.rpath.com>). In order to configure the rPath appliance, click the **System Configuration** link, located at the upper right corner of your browser work area; the following screen should be observed:



The screenshot shows the AsteriskNOW Sign In interface. At the top is the AsteriskNOW logo. Below it is a 'Sign In' header. The main area contains a 'User Name:' text box, a 'Password:' text box, and a 'Sign In' button. To the right of the password box are links for 'Schedule' and 'Updates'. The footer includes a 'Help' link, 'rPath Appliance Platform 2.0.3', and a copyright notice: 'Copyright© 2006-2007 rPath. All Rights Reserved.' with a 'Top of Page' link.

The username and password combination used to access the rPath configuration GUI is:

Username: admin

Password: password

Fill the appropriate text boxes, and click the **Sign in** button. The following screen should be observed:



The screenshot shows the AsteriskNOW Configuration Wizard. At the top is the AsteriskNOW logo. Below it is a header for 'rPath Appliance Platform Configuration Wizard'. On the left is a sidebar with links: 'User Management', 'Configure Email', 'Configure Notification', 'Configure Networking', and 'Backup and Restore'. The main area contains instructions: 'Add or update configuration information in the following pages. Click OK to save information for the page, and advance to the next configuration page. If the information does not require updating, click OK to continue on to the next page.' and an 'OK' button. To the right is a 'Welcome admin. Logout' message and links for 'Schedule' and 'Updates'. The footer includes a 'Help' link and a 'Top of Page' link.

Press the **OK** button to indicate to the appliance that you are ready. Once you have clicked the **OK** button, the screen will change enabling you to change the password for the rPath configuration GUI.



The password for the rPath configuration GUI and the password to the AsteriskNOW GUI are not the same. If you are installing the PBX for someone else to manage, don't set the same as the AsteriskNOW configuration GUI password.

Once you have set your new password, the wizard will continue to configure the various aspects of the appliance. Some of the settings include the configuration of a mail relay server for the appliance to send email notification with (if you have none, simply fill the form with the IP address 127.0.0.1), people to notify by email, and the backup methodology of the appliance.

If you are not a seasoned Linux sys-admin, the rPath appliance will make the task of managing your appliance easy. If you are a seasoned admin, the appliance may seem a little annoying to you at the start; however, you will find it highly useful for managing a remotely located PBX system. For more information about the rPath distribution, please refer to the rPath website at <http://www.rpath.com>.



Most experienced Linux sys-admins will try to log on to the AsteriskNOW appliance as root, only to find out they are unable to. The reason for this is that the root password is not configured during the installation, preventing a user from logging on as root to the appliance.

In order to log on as root, use the rPath configuration GUI to change the root password, via the **Configuration | Root Password** menu option.

Summary

In this chapter you successfully installed and executed the AsteriskNOW appliance and its initial configuration. You also set up two extensions – analog and VoIP – to be used with your PBX. You've set up our analog service provider, which is now ready to be used.

3

Extensions, Phones, and Others

Consider the following example, which is a conversation between a customer and an anonymous telephony technical support officer.

Junior Manager: Hello, I seem to be having issues making calls from my desk phone. Can you help me?

Tech Support Officer: OK sir, can you please tell me what kind of phone you have on your desk?

Junior Manager: Ahhh... a white one!

OK, the above case may sound a little drastic for a support scenario, however, such examples exist. While the concept of telephony is well known, to the non-techs, the operational concepts of the IP phone are a bit more complex. Before you continue with this chapter, you'll need to establish a few base lines.

An IP Phone is a Simplified Computer

Taking into consideration the fact that an IP phone implements both a telephone interface and an IP interface, it is clear that an IP phone is actually a computer with a simplified human interface.

This fact constitutes a mix-up, when non-technical people refer to an IP PBX system. Traditional PBX systems implement all the PBX services at the PBX level, while IP PBX systems tend to implement some of the features at the IP Phone level. For example, even the simplest IP phone is capable of rendering hold and transfer functions, without performing any function on the PBX system. This distinct difference must be taken into consideration in the design of your IP PBX system and most importantly, in the support your IP PBX environment.

Your choice of IP phone for your IP PBX environment is highly important. Each IP phone available in the market comes with a different feature set and reliability. Sometimes, the most costly phone isn't the answer and at times choosing a free or cheap phone for generic use may prove to be much more expensive on the support side.

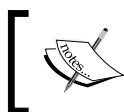
These facts must be taken into consideration, so that your installation is successful and users happy.

AsteriskNOW Extension Management GUI

Log on to your AsteriskNOW web management GUI, and click the **Users** link located at the left side of your screen – the main menu. The following screen should appear on your screen:

The screenshot displays the AsteriskNOW web management interface for user configuration. On the left, a sidebar menu lists various system functions, with 'Users' currently selected. The main content area is titled 'User and Phone Configuration' and features a 'User Extensions' list on the left, showing extensions 6000, 8500, and 8700 with their respective descriptions. To the right of this list are form fields for configuring a user, including Extension, Name, Password, VM Password, E-mail, Caller ID, Analog Phone, and Dial Plan. Below these fields is a section for 'Extension Options' with checkboxes for various features like Voicemail, SIP, CTI, Call Waiting, Can Reinvite, In Directory, IAX, Is Agent (checked), 3-Way Calling, and NAT. A 'DTMFMode' dropdown is set to 'rfc2833'. At the bottom of the main area are 'Save' and 'Cancel' buttons. On the far right, there is a 'Calling Rule' section with a note about choosing a rule plan, and a 'BuyNow' button for purchasing a phone. The footer contains copyright information for Digium, Inc. from 2006-2007.

As you can see, the screen is split into three distinct columns: the **User Extensions** list on the left, the options located in the middle, and an online quick help located at the right-most side of the screen.



This GUI methodology is unified across all of the AsteriskNOW web GUI, so the methodology of the GUI will not be explained in future chapters.

To create a new user extension, simply click the **New** button, located below the **User Extensions** list box. The following should appear on your screen:

User Extensions:

- 6000 -- Nir Simionovich
- 8500 -- Check Voicemail
- 8700 -- Fax
- New Entry**

Extension: 6001

Name: New User

Password:

VM Password: 1234

E-mail:

Caller ID:

Analog Phone:

Dial Plan:

Extension Options:

- ☒ Voicemail
- ☒ SIP
- ☐ CTI
- ☒ Call Waiting
- ☐ Can Reinvite
- ☐ In Directory
- ☒ IAX
- ☒ Is Agent
- ☒ 3-Way Calling
- ☐ NAT

DTMFMode: rfc2833

Codes

Call Waiting: Check this option if the User or Phone should have Call-Waiting capability.

Need a phone?
Click on the BuyNow button to purchase a phone directly from the Asterisk Configuration GUI.

Buttons: New, Delete, Save, Cancel, BuyNow

Unlike in Asterisk itself (from this point onwards, the compiled version of Asterisk will be referred to as "vanilla asterisk"), the definition of a new user extension is a fairly simple task. Fill out the various options of the provisioning interface, thus, at the end creating a new SIP-based extension for your PBX system.

The User Extensions Configuration Options

Option	Description	Value Type	Example
Extension	The extension number to be created. This field is usually automatically assigned, according to the last created extension.	Integer	6501
Name	The name of the person assigned to this extension.	Alpha Numeric	John Doe
Password	The device password assigned to this extension.	Alpha Numeric	Rft5yhbvy7
VM-Password	The initial password assigned for the extension voicemail mailbox.	Integer	123456
E-Mail	The email password of the person assigned to this extension, used for voicemail to email forwarding.	Alpha Numeric	john@doe.com
Caller ID	The outbound caller ID assigned to the user extension, which is visible to others on outbound calls from the device.	Alpha Numeric	800-500-1234
Analog Phone	The FXS Zap channel assigned for this extension – if any.	Combo	Analog Port #4
Dial Plan	The dial plan logic to be invoked for calls originating from this device.	Combo	DialPlan1



Please take note that at this point you don't have knowledge of how a dial plan works. Please don't be alarmed at this; the empirical knowledge of what a dial plan is will suffice at this point in time.

The "User Extension" Configuration Flags

Option	Description
Voicemail	Should your user have a voicemail mailbox?
SIP	Should your device be a SIP-based device?
CTI	Should your user be defined with Asterisk Manager login abilities?
Call Waiting	Is your device capable of rendering call waiting features?
Call Reinvite	Is your device capable of rendering SIP re-invite features? (Applicable to SIP devices only!)
In Directory	Should your provisioned user be listed in the PBX directory?
IAX	Should your device be an IAX device?

Option	Description
3-Way Calling	Is your device capable of rendering 3-way calling features?
Is Agent	Will this device or user be part of a calls queue?
NAT	Is your device located behind a NAT-Firewall or NAT router? (Applicable to SIP devices only!)
DTMF Mode	What type of DTMF signalling is your device using: • IN-BAND • RFC2833 • INFO (Applicable to SIP devices only!)

The configuration of your device is highly dependent on the vendor of your device. While a specific vendor may introduce a certain feature set, some other vendor may introduce another – resulting in dissimilar results for what would seem identical hardware phones or soft phones.

You will now create three new user extensions, based upon the SIP signalling protocol. The configuration should be in accordance to the following:

Device	Options	Flag
Device: 6510	Extension: 6510 Name: John Doe Password: 6510 VM-Password: 1234 e-Mail: john1@doe.com Caller-ID: 6510 Dialplan: DialPlan1	Voicemail: Yes SIP: Yes CTI: No Call Waiting: Yes Call Re-invite: Yes In Directory: Yes IAX: No Is-Agent: No 3-Way Calling: Yes NAT: No DTMF Mode: RFC2833
Device: 6511	Extension: 6511 Name: John Doe 2 Password: 6511 VM-Password: 1234 e-Mail: john2@doe.com Caller-ID: 6511 Dialplan: DialPlan1	Voicemail: Yes SIP: Yes CTI: No Call Waiting: Yes Call Re-invite: Yes In Directory: Yes IAX: No Is-Agent: No 3-Way Calling: Yes NAT: No DTMF Mode: RFC2833

Device	Options	Flag
Device: 6512	Extension: 6512 Name: John Doe 3 Password: 6512 VM-Password: 1234 e-Mail: john3@doe.com Caller-ID: 6512 Dialplan: DialPlan1	Voicemail: Yes SIP: Yes CTI: No Call Waiting: Yes Call Re-invite: Yes In Directory: Yes IAX: No Is-Agent: No 3-Way Calling: Yes NAT: No DTMF Mode: RFC2833

In addition to the above SIP phones, please configure an analog extension for your office fax machine. Please note that your FXS analog port is located at Analog Port #4. The configuration should be as follows:

Device	Options	Flag
Device: 6600	Extension: 6600 Name: Fax Machine Password: 6600 VM-Password: e-Mail: Caller-ID: 6600 Dialplan: DialPlan1 Analog Phone: Analog Port #4	Voicemail: No SIP: No CTI: No Call Waiting: No Call Re-invite: No In Directory: No IAX: No Is-Agent: No 3-Way Calling: No NAT: No DTMF Mode: RFC2833

At this point, your PBX is now ready to accept a few user extensions. Go ahead and connect your fax machine to the 4th port on the TDM card; if all goes well, you should hear a constant dial tone. The configuration interface of a popular IP phone, the LinkSys 941 IP Phone will now be introduced.

The LinkSys 941

The LinkSys 941 is one of the most popular hardware IP phones in the market, and you can obtain it via the Internet (prices vary from 120\$ to 180\$, depending on your location in the world).



Important Note: Most IP phones, including the LinkSys 941 and other LinkSys phones provide a means of automatic centralized provisioning and control, via the usage of a TFTP or HTTP server. For additional information about utilizing these configuration tools, please refer to each of the respective vendors websites.

The LinkSys 941 includes a web-based configuration GUI, which can be accessed directly from your computer's web browser. To find out with what IP number your IP phone is configured, please refer to your IP phone manual. Once you have logged on to your IP phone configuration GUI, the following main configuration summary should appear on in your web browser.

LINKSYS®
A Division of Cisco Systems, Inc.

Linksys Telephone Configuration

Info | System | User

[Admin Login](#) | [basic](#) | [advanced](#)
[Personal Directory](#) | [Call History](#)

System Information			
DHCP:	Enabled	Current IP:	192.168.2.106
Host Name:	SipuraSPA	Domain:	
Current Netmask:	255.255.255.0	Current Gateway:	192.168.2.1
Primary DNS:	192.168.2.1		
Secondary DNS:			
Product Information			
Product Name:	SPA-941	Serial Number:	88014FA21337
Software Version:	5.1.5	Hardware Version:	1.0.0(8830)
MAC Address:	000E08DBE33F	Client Certificate:	Installed
Licenses:	None		
Phone Status			
Current Time:	7/29/2007 06:34:07	Elapsed Time:	00:11:55
Broadcast Pkts Sent:	2	Broadcast Bytes Sent:	684
Broadcast Pkts Recv:	973	Broadcast Bytes Recv:	202444
Broadcast Pkts Dropped:	0	Broadcast Bytes Dropped:	0
RTP Packets Sent:	0	RTP Bytes Sent:	0
RTP Packets Recv:	0	RTP Bytes Recv:	0
SIP Messages Sent:	0	SIP Bytes Sent:	0
SIP Messages Recv:	0	SIP Bytes Recv:	0
External IP:			

Note that you are currently located at the information page. In order to start configuring your IP phone, click the **Admin Login** link and then the **advanced** link, both located at the upper right corner of your information page. The following screen should be observed:



LINKSYS® A Division of Cisco Systems, Inc.		Linksys Telephone Configuration	
Info	System	SIP	Provisioning
	Regional	Phone	Ext 1
	Ext 2	User	
		User Login basic advanced Personal Directory Call History	
System Information			
DHCP:	Enabled	Current IP:	192.168.2.106
Host Name:	SipuraSPA	Domain:	
Current Netmask:	255.255.255.0	Current Gateway:	192.168.2.1
Primary DNS:	192.168.2.1		
Secondary DNS:			
Product Information			
Product Name:	SPA-941	Serial Number:	88014FA21337
Software Version:	5.1.5	Hardware Version:	1.0.0(8830)
MAC Address:	000E08DBE33F	Client Certificate:	Installed
Licenses:	None		

Please note that the above is just an extract and not the entire screen. The important part in this extract is the top menu, providing the configuration of **Ext 1** and **Ext 2**. **Ext 1** and **Ext 2** represent our IP phone user accounts, which means that you can either connect your IP phone to two PBX systems or utilize two different device accounts on the same PBX system.

Click the **Ext 1** link, to bring up the configuration pages for the first device account. The following screen should appear on your screen:

The screenshot shows the Linksys Telephone Configuration interface. At the top, the Linksys logo is on the left, and "Linksys Telephone Configuration" is on the right. Below the logo, it says "A Division of Cisco Systems, Inc.". A navigation bar contains tabs: Info, System, SIP, Provisioning, Regional, Phone, Ext 1 (selected), Ext 2, and User. To the right of the tabs are links: User Login, basic, advanced, Personal Directory, and Call History. The main content area is titled "General" and contains several sections: "Line Enable:" with a dropdown set to "yes"; "Share Line Appearance" with "Share Ext:" set to "private", "Subscription Expires:" set to "3600", and a "Shared User ID:" field; "NAT Settings" with "NAT Mapping Enable:" set to "no", "NAT Keep Alive Enable:" set to "no", "NAT Keep Alive Msg:" set to "\$NOTIFY", and "NAT Keep Alive Dest:" set to "\$PROXY"; and "Network Settings" with "SIP TOS/DiffServ Value:" set to "0x68", "SIP CoS Value:" set to "3", "RTP TOS/DiffServ Value:" set to "0xb8", "RTP CoS Value:" set to "6", "Network Jitter Level:" set to "high", and "Jitter Buffer Adjustment:" set to "up and down".

Scroll down the page to locate the following section:

The screenshot shows two sections of the configuration page. The first section is "Proxy and Registration" and contains: "Proxy:" field, "Outbound Proxy:" field, "Register:" set to "yes", "Register Expires:" set to "3600", "Use DNS SRV:" set to "no", "Proxy Fallback Intvl:" set to "3600", "Use Outbound Proxy:" set to "no", "Use OB Proxy In Dialog:" set to "yes", "Make Call Without Reg:" set to "no", "Ans Call Without Reg:" set to "no", "DNS SRV Auto Prefix:" set to "no", and "Proxy Redundancy Method:" set to "Normal". The second section is "Subscriber Information" and contains: "Display Name:" field, "Password:" field, "Auth ID:" field, "Mini Certificate:" field, "SRTP Private Key:" field, "User ID:" field, and "Use Auth ID:" set to "no".

The above section enables you to configure your device SIP information, to be used in conjunction with the information we've provisioned using your AsteriskNOW web configuration GUI. You will now configure the user extension designated 6500, in accordance with the information you configured before (the relevant configuration parameters are listed in the following table):



The AsteriskNOW installation used for the writing of this book is configured with the IP address 192.168.2.122. Your installation may differ.

Parameter	Description	Value
Proxy	The IP address or domain name of your SIP Proxy server – in our case, the IP address of the AsteriskNOW PBX.	192.168.2.122
Outbound Proxy	The IP address or domain name of your outbound SIP Proxy server – in our case, the IP address of the AsteriskNOW PBX.	Leave blank
Register	Does your SIP Proxy require registration? (default: Yes)	Yes
Register Expire	How long is a registration to the SIP proxy is valid, in seconds? (default: 3600)	3600
Use DNS SRV	Should the IP phone utilize DNS resolving for SIP communications? (default: No)	No
Use Outbound Proxy	Should the IP phone utilize the Outbound Proxy setting for the media stream (default: No).	No
Display Name	The device's display name to be added to the caller-ID.	6500
Auth-ID	The device's authentication user name to be used when registering to the SIP Proxy.	6500
Password	The device's authentication password to be used when registering to the SIP Proxy.	6500
Use Auth-ID	If set to Yes the username field for authentication will be taken from the Auth-ID field, otherwise, the User-ID is used (default: No).	Yes
User-ID	The device's user ID, usually a fully qualified telephone number.	6500

Once you have completed setting up your IP phone, and clicked the **Submit All Changes** button, located at the bottom of the configuration GUI, your IP phone will be rebooted.

After a successful reboot, your phone will try to register with the AsteriskNOW PBX, if all goes well, all four green lights should appear on the front panel of your IP phone. It should look as follows:



If your phone looks like the above screenshot, congratulations! You have a working IP phone registered with your AsteriskNOW PBX system. We shall now see how to configure an IP-based soft phone application.

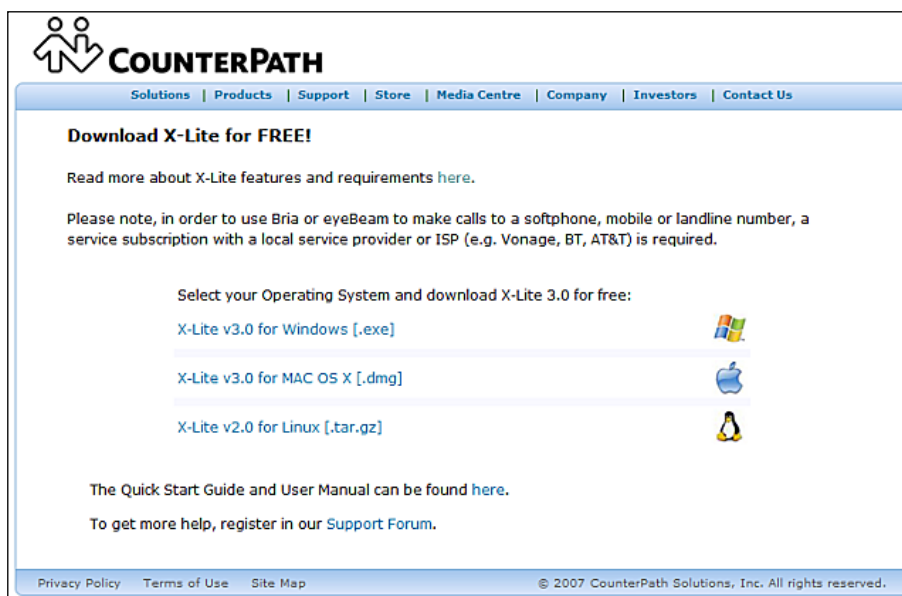
CounterPath X-Lite—The Worlds Most Popular Soft Phone

If you were to ask almost any VoIP engineer what is the best IP phone they can think of, the answer you would get would most probably be CounterPath. CounterPath was known in as Xten—makers of X-Lite and X-Pro—however, with the introduction of eyeBeam (their SIP Video Phone) they changed the name to CounterPath.

You can download the CounterPath X-Lite SIP phone from the following website:

<http://www.counterpath.com/index.php?menu=download>

Upon loading the website, the following screen should appear:



Please note that X-Lite comes in several versions, each one for its own operating system. So make sure you download the version applicable to your operating system. Once you finish your download and installation (please note that the installation performs a test of your microphone and speakers, so have those handy for the installation), the following screen should appear on your monitor:



Press the right mouse button, and select the SIP accounts settings option. A list box will appear, indicating that there are no SIP accounts enabled. Press the **Add** button; the following screen should be observed:

The screenshot shows a Windows-style dialog box titled "Properties of Account1". It has five tabs: "Account", "Voicemail", "Topology", "Presence", and "Advanced". The "Account" tab is selected. Inside the "Account" tab, there are two main sections. The first section, "User Details", contains five text input fields: "Display Name", "User name", "Password", "Authorization user name", and "Domain". The second section, "Domain Proxy", contains a checked checkbox labeled "Register with domain and receive incoming calls", followed by the text "Send outbound via:". Below this text are three radio button options: "domain", "proxy", and "target domain". The "target domain" option is selected. To the right of the "proxy" radio button is a text input field labeled "Address". At the bottom of the dialog, there is a "Dialing plan" label and a text input field containing the string "#1\|a\|a.T;match=1;prestrip=2;". At the very bottom of the dialog are three buttons: "OK", "Cancel", and "Apply".

You are now required to configure the SIP phone in order to utilize your AsteriskNOW configuration. Fill out the following information in the window:

Display Name	6501
User Name	6501
Password	6501
Authorization user name	6501
Domain	192.168.2.122
Register with domain and receive incoming calls	Enable
Send outbound via	domain

If all goes well, you should now be able to make calls from your configured IP phone to your configured soft phone. In addition, if you have connected your fax machine to the FXS interface, and if you call your fax machine extension (extension 6600), your fax machine should ring and answer the call.



NAT/PAT configurations

NAT (Network Address Translation) and PAT (Port Address Translation) don't mix well with the SIP signalling protocol; this is caused by the fairly simplistic mechanism of SIP. While AsteriskNOW enables the marking of a SIP user device to support NAT, it is necessary to validate what is the proper NAT configuration for your IP phone. Additional configuration information about NAT and your IP phone can be found at the website of your respective IP phone maker.

Summary

In this chapter you successfully defined and configured three user extensions to be used with our AsteriskNOW PBX system. You successfully tested the traversal of calls from one IP phone to the other, and from the IP phones to the fax extension (located on an FXS interface).



If you run into issues during any chapter, you can contact me via email at: asterisknowbook@greenfieldtech.net, or via my AsteriskNOW support forum at <http://www.greenfieldtech.net/support/>.

4

Service Providers—Your Connection to the World

I try to do the right thing with money. Save a dollar here and there, clip some coupons. Buy ten gold chains instead of 20. Four summer homes instead of eight. — L. L. Cool J

In the traditional telephony world, a service provider is a business entity that provides you with analog connection to the PSTN, assigns you a phone number, and remembers to send you a bill at the end of every month. In that respect, a VoIP carrier is very much similar to a traditional one, except that the technology is different. However, due to the nature of IP technologies, a radical difference exists—you are no longer limited to a single carrier.



While in some countries it is fairly common to enjoy PSTN connections from several carriers at the same time, in other countries this possibility doesn't exist.

VoIP Carriers

The IP telephony world is made of two types of carriers, DID (Direct Inward Dialing) carriers and termination carriers. Before you continue with this chapter, take some time to understand these terms.

Direct Inward Dialing (DID/DDI) Carriers

As the name suggests, a DID carrier provides a phone number to be connected to your PBX via an IP transit. Most DID providers provide DID numbers in more than a single region of the world, thus, enabling a single IP PBX to enjoy multiple incoming DID numbers.

DID providers are split into two sectors – the carrier/wholesale market and the retail market. Providers focused on the carrier/wholesale market operate under the assumption that their customers want to buy and sell DIDs in various regions of the world. Thus, their operational environment operates as a multi-carrier switching environment. A typical example of such a carrier is DIDx (<http://www.didx.net>). DIDx provides a DID exchange platform for carriers, enabling them to easily connect to one another, and exchange DIDs. This is how a carrier in the US can easily obtain DID numbers in places like Iraq and re-sell those in the US.

Providers focused on the retail market operate under the assumption that their customers are looking for a way to connect multiple DID numbers across the world, into a single IP PBX installation. One of the biggest DID providers in the retail arena (if not the biggest in the world) is VoxBone (<http://www.voxbone.com>). VoxBone provides customers with a very simple web GUI, enabling them to purchase DID numbers in various regions of the world and then directing them to their PBX via SIP. Another retail DID provider is LibréTel (<http://www.libretel.com>), which provides a fairly similar service.



If you want to seek additional DID providers, you would most probably find others. The main problem with a DID provider is – if your DID provider isn't a well established company, and is a small operation, then you may end up with a provider that buys and sells DID numbers from other companies. Now, if that provider isn't able to carry out its business properly, then you may end up with no service. Take time to select your DID providers, study them, evaluate them, and most importantly, make sure you have a backup.

IP Call Termination Carriers

As the name suggests, call termination carriers are providers rendering services to connect the IP worlds and the traditional PSTN world. You may be familiar with the term PC-to-Phone – a service enabling voice calls from a PC to a normal (landline or mobile) phone. The term "termination" is based upon the idea that a termination carrier is the last point before the call leaves the IP world and enters the PSTN world – thus the terminating point in the infrastructure. Like their DID counterparts, call termination providers also operate in two sectors, the carrier/wholesale sector and the retail sector.

Providers focused on the carrier/wholesale sector operate under the assumption that their customers are other carriers or wholesale users (calling card operators, call shops, service providers, etc.) who are able to generate a high volume of calls (millions of minutes) per day. A typical example for a carrier/wholesale provider would be a company like iBasis (<http://www.ibasis.com>). iBasis provides termination services for carriers, with the ability to terminate calls to multiple parts of the world, while keeping a high service level.

Providers operating in the retail sector operate under the assumption that their customers are end users, or in the high-end enterprises that are able to generate a moderate volume of calls (thousands to tens of thousands of minutes) per month. A typical example for call termination partners in the retail sector would be Vonage (<http://www.vonage.com>). Vonage has positioned itself as a residential/business-oriented VoIP carrier, rendering services world wide. Based upon a flat rate fee billing model, Vonage is a highly popular provider. In addition to being a termination partner, Vonage also provides DID numbers for its customers in the US.

An alternative to Vonage, which is a world-wide carrier, would be NuFone (<http://www.nufone.net>). NuFone is a regional VoIP carrier in the USA, operating mainly in the state of Michigan. While NuFone is much smaller than Vonage, its ability to be flexible with its customer is of high value; thus, if your business is located in Michigan, NuFone may be a better choice for you.

Refilers and Grey Routes

Like any other business, the VoIP carrier market is infested by various so-called carriers, which are not carriers at all. These are referred to as **refilers** (at the left side of the spectrum) and **grey routes** (at the right side of the spectrum).

A refiler is actually a person who has the ability to connect with multiple carriers, and then provide a single entry point to these carriers. In general, a refiler aggregates call traffic from multiple points, usually small, into several large scale carriers.

Grey route providers are the bad boys of the VoIP business. They are capable of establishing a route to a certain location, operating it for a certain period of time, shutting it down for some reason, and then disappearing with your money. A large number of refilers around the world work with multiple grey route providers, thus, aggregating multiple grey routes. You must have now realized that connecting with a refiler may not be the right thing for you, if you intend to operate a proper business. If you do intend to use refilers, make sure you have a few of them working for you, so that you don't get stuck.



If you have a doubt about your choice of termination partner, visit the voipbadguys website (<http://www.voipbadguys.com>). While the site is a slightly old and outdated, the information available may save your money (it had saved me in the past a couple of times). If you've encountered a carrier that had breached your trust, or you suspect them of wrong doings, report them to voipbadguys.com, and include them in the list.

PSTN Carriers—Traditional Telephony Providers

While AsteriskNOW is an IP PBX environment, it is still capable of connecting with traditional PSTN carriers. This connection is performed using connectivity hardware (which can be purchased from Digium, the makers of AsteriskNOW).

PSTN carriers provide connectivity to your PBX utilizing any of the following interfaces—Primary Rate Interfaces (PRI), Basic Rate Interfaces (BRI), and Foreign Exchange Office Interfaces (FXO).

PSTN carriers provide both an inbound service (DID provider) and an outbound service (termination provider) utilizing the same physical link to the PSTN network. While PRI and BRI interfaces are fairly similar in technical capabilities (apart from their size), there is a cardinal difference between PRI/BRI interfaces and FXO interfaces from the user's point of view (for a telecom engineer it's a totally different case). While PRI/BRI interfaces are able to utilize DID provisioning, which means that when a call is intercepted by a PRI/BRI interface, it reports back to the PBX the DID that was called—an FXO interface is unable to do so. Inbound call routing using FXO interfaces is purely a physical operation, unlike VoIP and PRI/BRI where it is a logical operation.



Your AsteriskNOW installation will identify all your installed hardware upon installation; thus, you will not have to perform any special Operating System-level configuration to get your AsteriskNOW installation up and running.

If you may recall, your analog interfaces (FXO) have already been provisioned and configured in Chapter 2.

Configuring an IP Termination Service Provider

You have already provisioned the analog service provider connections of your PBX system; now configure an IP-based service provider for call termination services.

When connecting to a VoIP carrier, you will be assigned a username and password, clearly identifying your system at the service provider, and the IP number of the termination gateway (or FQDN). While this is a norm with most carriers, some carriers may provide only the IP (or FQDN) of the termination gateway. In such a case, you will have to install your PBX utilizing a valid, fully routable, IP address.



Routable IP addresses: IANA has provisioned a few classes of IP addresses to be utilized as internal networks. These IP classes are well documented and are considered to be "non-routable" within the IP space of the Internet. Following is a list of these classes; if a PBX is allocated an IP address within one of these classes, the ISP or system administrator needs to be contacted for a routable IP address.

Class A: 10.0.0.0 up to 10.255.255.255, Netmask: 255.0.0.0

Class B: 192.168.0.0 up to 192.168.255.255, Netmask: 255.255.0.0

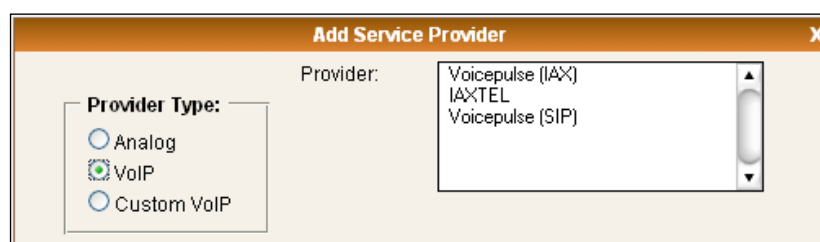
Class C: 172.16.0.0 up to 172.31.255.255, Netmask: 255.255.240.0

Recall that your AsteriskNOW installation is located at 192.168.1.122, which means that your PBX is located inside an internal network with a non-routable IP address. This means that the best way to connect to your service provider would be with a username/password combination.

VoIP Service Providers in AsteriskNOW

AsteriskNOW provides two types of VoIP service providers:

- Built-in VoIP service providers
- Custom VoIP service providers

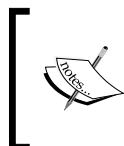


The built-in VoIP service providers include AsteriskNOW and Asterisk-friendly services providers, which have contributed in some form to the development or the distribution of Asterisk around the world. The idea of a built-in service provider is that the technical configuration is pre-configured within AsteriskNOW and the only thing remaining is to configure a username and a password.



Later in the book, you will discover how to add your own built-in VoIP Service Provider.

Once you have selected your preferred service provider the following screen will appear:



VoicePulse has been a long time Asterisk promoter, and its services are friendly to Asterisk users. Visit its website at <http://www.voicepulse.net>, for additional information about its services.

The **Custom VoIP** service provider option is slightly more complex, and requires some additional information, which is explained next.

Once you have selected **Custom VoIP** for **Provider Type**, the following screen will appear:

Add Service Provider

Provider Type:

☐ Analog

☐ VoIP

☒ Custom VoIP

Comment:

Protocol:

Register: ☒

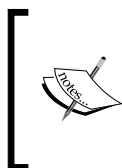
Host:

Username:

Password:

The following options are observed:

Option	Description	Example
Comment	A simple textual description for your new VoIP provider, which will be used later in routing information.	AsteriskNOW Book
Protocol	The signalling protocol designated for this service provider. Options are: SIP and IAX.	IAX
Register	Indicates if your service provider requires you to register in order to utilize the provider's service. Providers like Vonage and Packet8 may require registration, while termination service providers may not.	No
Host	The IP address of FQDN or the service provider's telephony server.	venus.greenfieldtech.net
Username	A username that was designated to you by the provider.	asterisknow
Password	A password that was designated to you by the provider.	asterisknow



Registration in the VoIP world isn't really a must, although, usually it makes life fairly simple. In a case where registration is not required by the provider, you might have to utilize a routable static IP address, a username/password combination, or both. It is best to consult the service provider's technical staff to ensure that the configuration is correct.

VoIP Service Providers—Few Examples

A few Asterisk-friendly service providers are introduced in this section. Please note that they are used as examples. They are not endorsed or recommended.

Inbound DID/DDI Service Providers

Inbound DID/DDI service providers provide a phone number for your IP PBX system. Adding phone numbers that are located in various regions of the world to its IP PBX is a crucial issue for a growing company. Imagine a start-up company being able to establish "virtual-offices" in various countries, without actually having anybody on location, especially when dealing with sales and support offices.

VoxBone—The Inbound VoIP Provider



VOXBONE
The inbound VoIP Provider

[Home](#)
[Services](#)
[Coverage](#)
[Partners](#)
[About us](#)
[Knowledge Base](#)
[Contact us](#)
[News](#)
[Sign Up](#)
[Login](#)

Login

Username :
Password :

Not a member yet? click [here to register](#)
[Forgot password ?](#)

[Terms And Agreements](#) - [Acceptable Use Policy](#)

voip origination - virtual numbers - did numbers - virtual phone numbers - sitemap - glossary

Copyright © 2005-2006, VOXBONE SA

VoxBone is one of the largest inbound VoIP providers in the world, rendering services in over twenty countries in the world. Its pricing model is fair and well adopted to the needs of its customers, and its network is one of the most stable VoIP networks in the world.

LibréTel—Freedom is Calling

LibréTel Freedom is calling. *It's for you.* a pulver.com company

Home Products & Services Sign Up Login FAQ

LibréTel, is a leading provider of customized IP communications services that leverage the entire communications spectrum:

- voice
- video
- presence
- instant messaging

Using our proven carrier-class platform, LibréTel supplies scalable, pervasive communications solutions to individuals and organizations worldwide.

User name:

Password:

[Forgot your password?](#)

Let your customers' clients, family and friends call them at a local number instead of paying international charges.

Give them virtual Libretel phone numbers from the U.S. or cities around the world.

Sign up for our reseller program today. Get these benefits for yourself now.

RESELLER

LEARN MORE **SIGN UP NOW!**

[Learn More About Our Direct Products and Services](#)

LibréTel is a pulver media company, which means that it's back by one of the world's biggest VoIP promoters—Jeff Pulver. Having an established service since 2004, LibréTel is a valid partner for your DID provisioning.

Termination and Residential Service Providers

Termination service providers provide you with a means to initiate calls from your IP PBX or IP phone and terminate the calls to a normal PSTN phone (or mobile phone). Residential providers are a combination of a DID provider and a termination provider, rendering both services at the same time.

Vonage

Vonage Select Your Country: United States **Already Saving with Vonage?** Username: Password: **GO** [Forgot your password?](#)

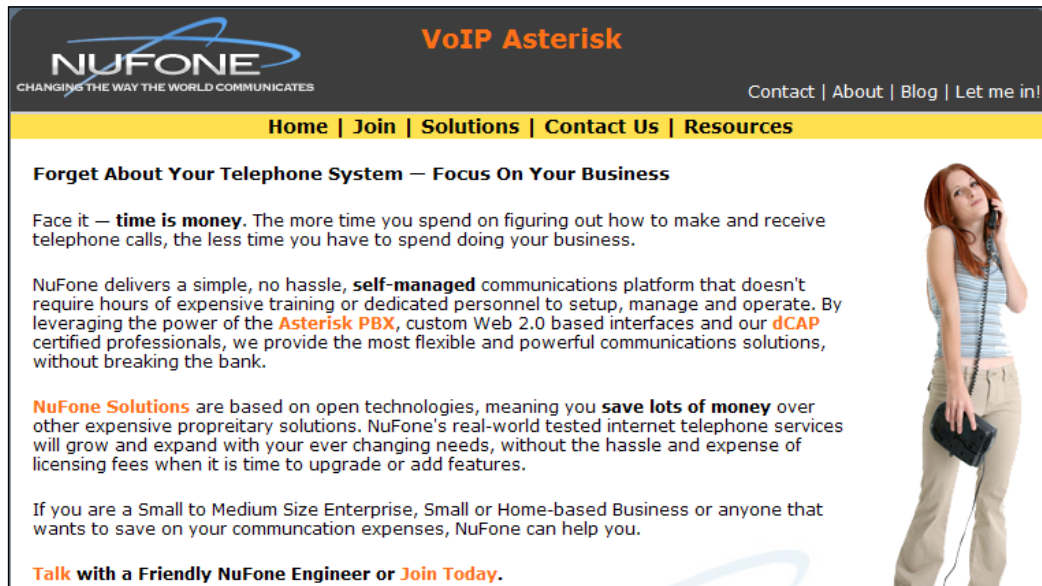
Home | Services | Products | Availability | Features | International Rates | Sign Up | Help Search

#1 Rated Digital Phone Service only **\$24.99/month** **UNLIMITED** local and long distance calling **30 day Money Back Guarantee**

Try Vonage 1 Month FREE! see details below **Sign Up Now!** **Or Call: 1-800-705-7092**

In the US, when people say VoIP, they usually mean Vonage. Vonage was one of first VoIP carriers in the world, and since then has constantly dominated the market. With its own IP network and a huge staff for technology and sales, Vonage is clearly a good choice for an outbound carrier.

NuFone—Changing the Way the World Communicates



NUFONE
CHANGING THE WAY THE WORLD COMMUNICATES

VoIP Asterisk

Contact | About | Blog | Let me in!

Home | Join | Solutions | Contact Us | Resources

Forget About Your Telephone System — Focus On Your Business

Face it — **time is money**. The more time you spend on figuring out how to make and receive telephone calls, the less time you have to spend doing your business.

NuFone delivers a simple, no hassle, **self-managed** communications platform that doesn't require hours of expensive training or dedicated personnel to setup, manage and operate. By leveraging the power of the **Asterisk PBX**, custom Web 2.0 based interfaces and our **dCAP** certified professionals, we provide the most flexible and powerful communications solutions, without breaking the bank.

NuFone Solutions are based on open technologies, meaning you **save lots of money** over other expensive proprietary solutions. NuFone's real-world tested internet telephone services will grow and expand with your ever changing needs, without the hassle and expense of licensing fees when it is time to upgrade or add features.

If you are a Small to Medium Size Enterprise, Small or Home-based Business or anyone that wants to save on your communication expenses, NuFone can help you.

Talk with a Friendly NuFone Engineer or Join Today.

NuFone is headed by Jeremy McNamara, one of the best known figures in the Asterisk community. Jeremy established NuFone back in 2004 to provide termination and residential services to Asterisk users. NuFone is dCAP certified, which means that if you're using Asterisk or AsteriskNOW, they will be able to support your Asterisk installation easily and get you up and running as fast as possible.

Connecting to a Custom VoIP Termination Provider

For the purpose of this demo, a SIP termination point has been created for you to configure. Once successfully configured, every call made to this VoIP provider will end up in an automated attendant system, which is located in my office.

So, now you need to create a Custom VoIP provider that will include the following configuration:

Option	Value
Comment	AsteriskNOW_Book
Protocol	IAX
Register	No
Host	venus.greenfieldtech.net
Username	asterisknow
Password	asterisknow

After configuring the service provider, you need to add an outbound rule to your routing table. Click the **Calling Rules** menu option; the routing table should look as follows:

S.No	RuleName	Dial Pattern	Call Using	Options
1	all_outbound	Begins with 9 and followed by 3 or more digits	Ports 1,2,3	Edit Delete

Click the **Add a Calling Rule** button to create a new routing rule, and fill in the following information below (don't worry, this will be explained in the next chapter).

Rule Name:

Place this call through :

Dialing Rules : If the number begins with and followed by digits ☒ or more
(define a custom pattern)

Strip digits from the front and prepend before dialing

This configuration indicates that your PBX will analyze any number dialled, and if the number begins with the digit **5** and has a minimum of **3** digits following it, the call will be routed into the Custom VoIP provider you created. Before routing the call to the provider, AsteriskNOW will remove the leading **5** from the number, and then forward the call to the termination provider.

Now, after creating the configuration, click the **Activate Changes** button, located at the top-right side of your browser screen. Now, pick up the phone and dial **512345**, you should be greeted with a recorded message.

Summary

In this chapter you configured a new VoIP termination service provider for your PBX, and also configured an initial routing table to enable the usage of this service provider. You should now have two service providers configured on your PBX – an analog service provider and a VoIP service provider.

5

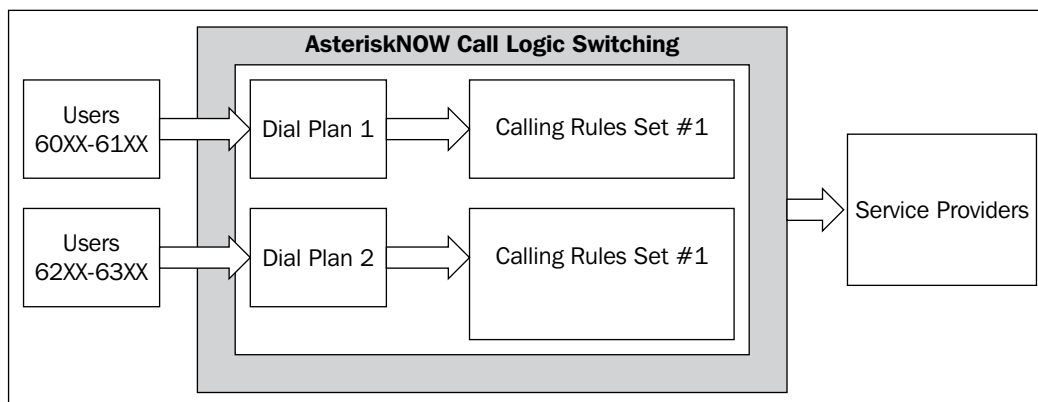
Tentacles of the PBX—The Calling Rules Tables

Philosophy: A route of many roads leading from nowhere to nothing.

— Ambrose Bierce

While it appears that Ambrose Bierce has a little respect for Philosophy in general, he would surely refer to calling routing as — roads leading from a single point to multiple points, or something similar. The purpose of the "Calling Rules" is to define the methodology by which calls are routed from the PBX to the outbound world.

An interesting feature of a PBX is the ability to support multiple dial plans, meaning that you are able to create various dial-plan logics, associate different calling rules to each dial plan, and assign users to specific dial plan (as seen in Chapter 3). Schematically speaking, the relation between the three can be illustrated as follows:



Essentially, from AsteriskNOW's point of view, any dial attempt that doesn't match a Calling Rule will be considered an internal call, and thus, AsteriskNOW will try to route the call to an internal AsteriskNOW resource – e.g. another extension or an AsteriskNOW feature code.



Most telephony engineers and carrier support engineers refer to "Calling Rules" as "Routing Rules". From this point onwards in the book "Calling Rules" will be referred to as "Routing Rules".

Managing Routing Rules with AsteriskNOW

At this point, after initially configuring your service providers and initial routing rules, your "Routing Rules" table should look as follows:

S.No	RuleName	Dial Pattern	Call Using	Options
1	all_outbound	Begins with 9 and followed by 3 or more digits	Ports 1,2,3	Edit Delete
2	AsterisNOW_Book	Begins with 5 and followed by 3 or more digits	Custom - AsteriskNOW_Book	Edit Delete

Edit one of these rules to get acquainted with the call rule dialog box. Click the **Edit** link of the **all_outbound** rule (rule 1). The following dialog box should appear on your screen:

Rule Name:

Place this call through :

Dialing Rules : If the number begins with and followed by digits ☒ or more
(define a custom pattern)

Strip digits from the front and prepend before dialing

Every call made from an IP phone connected to the PBX is processed by the routing rules. The processing is performed in the following order:

- AsteriskNOW grabs the dialled number and tries to match it to the prefix defined in the Routing Rule. In this dialog, the prefix is 9.
- It then verifies the number of digits suffixing the prefix. In this example, any number of digits that is 3 or more is considered a valid number to be assigned to this route.
- Now, before actually routing the call to the designated service provider, AsteriskNOW can remove prefixes and/or prefix numbers to the dialled number. In this example it will only remove a single prefixing digit (9) and then pass the call to your service provider—**Ports 1,2,3**.

The above process happens for every call that is made by a phone connected to the PBX. If the process fails all the rules defined in the Routing Table, AsteriskNOW assumes that the call is supposed to be routed internally. If internal routing fails, the call will fail and a fast-busy tone will be heard from your IP phone.



Some IP phones also indicate the SIP error message that was received. If routing fails, the normal error that you may encounter would be error 404 – NOT FOUND.

Manually Editing Dial-Plan Logic

CAUTION!

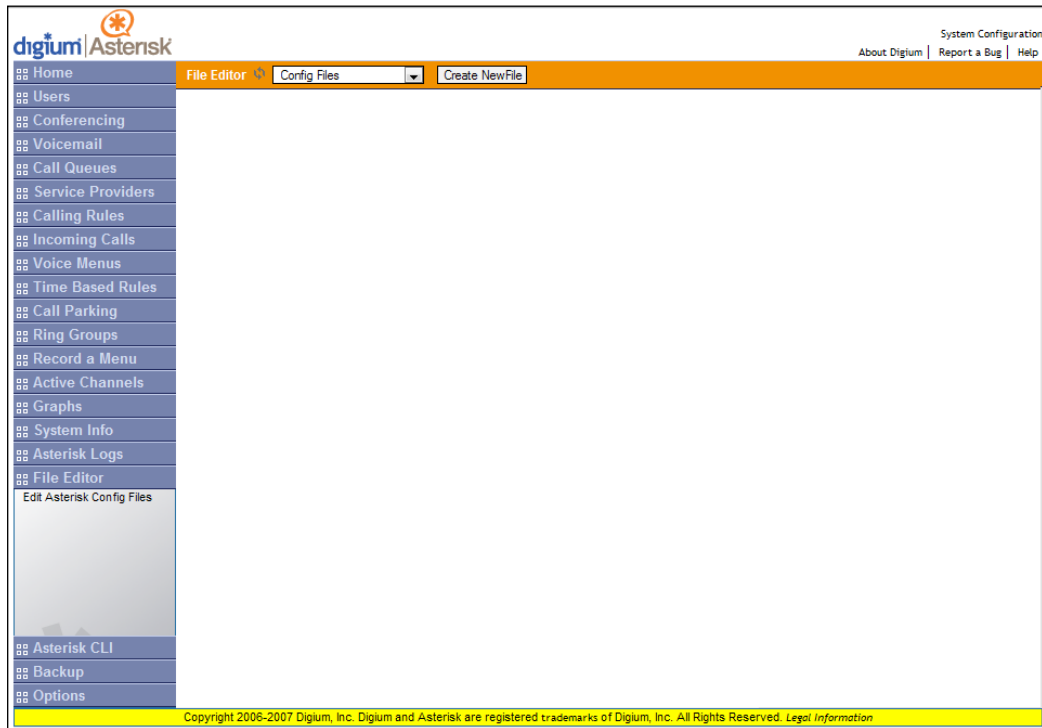
The following section is for advanced users only! If you are not familiar with programming languages, at this point, please skip to the next chapter.



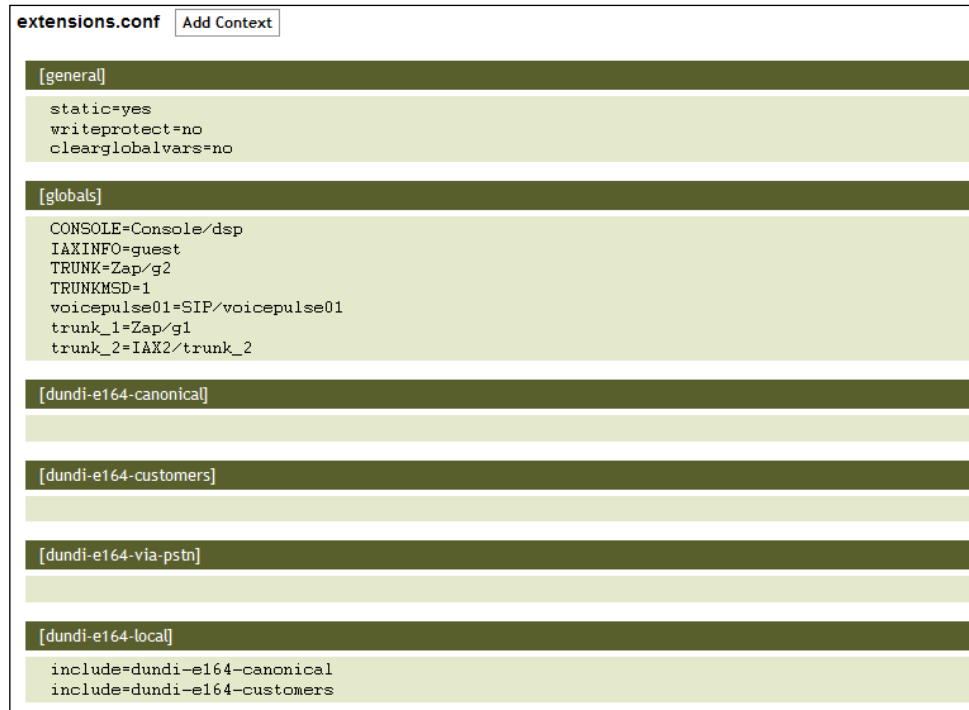
Additional information about Asterisk configuration files

While this book deals with AsteriskNOW – aimed to provide a simple, fast solution for managing the Asterisk Open Source PBX – it does so without getting into the inner workings of Asterisk's configuration files. Additional information about the configuration files, their format, usage, and various available options is available at the voip-info wiki, located at: <http://www.voip-info.org/wiki/view/Asterisk+config+files>.

Manually editing AsteriskNOW dial-plan files isn't recommended; however, in some extreme situations it can't be avoided (especially while creating special applications with AsteriskNOW). For this specific reason, AsteriskNOW includes a facility to edit the configuration files manually. Click the **File Editor** menu option; the following screen should appear:



The `extensions.conf` file should interest you. Using the drop-down file selector, select the **extensions.conf** file for editing. The following screen should be observed:



Now, if you were to click one of the lightly-shaded areas (light green on your screen); a text editing box will open to enable you to edit the file.

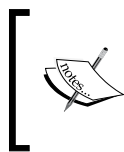


You're probably wondering at this point – "where is routing logic located?" Scroll down the file and seek the section designated as **numberplan-custom-1**, which designates your **DialPlan1** dial plan. If you were to create an additional dial plan using the GUI, its designation in the `extensions.conf` file would be **numberplan-custom-2**.

Examine the following section:

```
[numberplan-custom-1]
plancomment=DialPlan1
include=default
exten=_9XXX!,1,Macro(trunkdial,${trunk_1}/${EXTEN:1})
comment=_9XXX!,1,all_outbound,standard
exten=_5XXX!,1,Macro(trunkdial,${trunk_2}/${EXTEN:1})
comment=_5XXX!,1,AsterisNOW_Book,standard
```

In the above section lines indicated by the **exten** directive, indicate a dial-plan activity. The **exten** directive is then followed by some form of well formatted number string, followed by a number indicating the sequence number of the directive, then followed by an Asterisk operational command – in our case, the **Macro** command.



Asterisk and AsteriskNOW include over 150 different applications; explanation of each and every application is beyond the scope of this book. Visit the Asterisk Wiki page at <http://www.voip-info.org/wiki/index.php?page=Asterisk> for more information about Asterisk and AsteriskNOW applications.

The way the dial plan analyzes dialed numbers is explained next. Consider one of the dialing rules:

```
exten=_9XXX!,1,Macro(trunkdial,${trunk_1}/${EXTEN:1})
```

The interesting portion of this line is the `_9XXX!`. The following is an extract from an Asterisk documentation:

```
; Extension names may be numbers, letters, or combinations
; thereof. If an extension name is prefixed by a '_'
; character, it is interpreted as a pattern rather than a
; literal. In patterns, some characters have special meanings:
;
; X - any digit from 0-9
; Z - any digit from 1-9
; N - any digit from 2-9
; [1235-9] - any digit in the brackets (in this example,
1,2,3,5,6,7,8,9)
; . - wildcard, matches anything remaining (e.g. _9011. matches
```

```

; anything starting with 9011 excluding 9011 itself)
; ! - wildcard, causes the matching process to complete as soon as
; it can unambiguously determine that no other matches are possible
;
; For example the extension _NXXXXXX would match normal 7 digit dialings,
; while _1NXXNXXXXXX would represent an area code plus phone number
; preceded by a one.
;
; Each step of an extension is ordered by priority, which must
; always start with 1 to be considered a valid extension. The priority
; "next" or "n" means the previous priority plus one, regardless of whether
; the previous priority was associated with the current extension or not.
; The priority "same" or "s" means the same as the previously specified
; priority, again regardless of whether the previous entry was for the
; same extension. Priorities may be immediately followed by a plus sign
; and another integer to add that amount (most useful with 's' or 'n').
; Priorities may then also have an alias, or label, in
; parenthesis after their name which can be used in goto situations

```

As this text suggest, the underscore marking (`_`) indicates the start of a pattern matching rule. This is then followed by a form of expression indicating the pattern to match. In the example, the pattern match is `_9xxx!`, so, interpreting this according to the documentation:

- `_9`: Indicates any number that is prefixed with the digit 9. This corresponds to the first routing rule.
- `xxx`: Indicates 3 digits, ranging from 0 to 9. This corresponds to the second portion of the routing rule.
- `!`: Indicates to match as soon as there is no other rule that may apply, thus closing the matching process.

For example, in accordance to the above points, the number 912345 will match the rule indicated above and will simply activate the Macro application. However, bearing in mind that the routing requires at least 3 digits to follow the prefix, the number 912 will not match our above routing rule.

At this point, the working of the Macro application is not explained; however, the empirical knowledge of its existence is enough. Chapter 11 deals with various advanced configurations with AsteriskNOW it covers the various applications and dial-plan workings.

Summary

In this chapter you learned what routing rules are and how they are processed within the AsteriskNOW operational model. Understanding the way routing rules work is imperative to configure your PBX for optimal usage of outbound connections.

6

"Let me in!"—Inbound Call Routing

*"Would you tell me, please, which way I ought to go from here?" asked Alice.
"That depends a good deal on where you want to get to," said the Cat.
"I don't much care where – " said Alice.
"Then it doesn't matter which way you go," said the Cat.
" – so long as I get somewhere," Alice added as an explanation. — Lewis Carroll,
Alice's Adventures in Wonderland.*

When a call is intercepted by a PBX system, its handling is very much like Alice's trips into wonderland – the call doesn't really know where it's going. Inbound call routing enables the interception of calls, determined by a preset routing rule-set, enabling various actions to be taken for different calls.

Inbound DID Routing versus Analog Physical Routing

Most PBX systems are capable of analyzing inbound calls based on two characteristics – the DID number the call was made to, and/or the physical analog port that intercepted the call.

Inbound Routing via DID Numbers

Inbound DID routing relies on the fact that the routing medium is capable of carrying the **called number** information to the PBX system. Such functionality exists in VoIP-based connections or digital connections (PRI/BRI). When the PBX receives a call, a certain element in the signalling protocol reports the **called number to the PBX**, which the PBX can analyze and act upon.

Your called number may vary depending on the service providers. For example, while service providers in some regions of the world send the full called number, others send only the last 6 digits of a called number. When working with a service provider, it is imperative that you fully understand the format your service provider uses to send you the information.

Inbound Routing via Physical Ports

Routing via physical ports is used when the connecting medium is unable to provide the PBX with the called number information. This situation is a norm when dealing with analog connections (FXO). The main reason for this is that a single analog connection is meant to have a single phone number assigned to it (so sending the called number parameter to the PBX is pointless).

Routing Type Comparison Table

The following table summarizes the inbound routing methodology used by each type of inbound connection.

Connection Type	DID Routing	Physical Routing
FXO	No	Yes
ISDN BRI	Yes	Yes
ISDN PRI	Yes	Available, although pointless
VoIP – SIP	Yes	No
VoIP – IAX2	Yes	No

Inbound Call Routing with AsteriskNOW

Recall that, during your initial configuration (Chapter 2), you configured your PBX to route all calls arriving at the analog ports to extension **6000**. Inbound calls can be routed to specific extensions, Interactive voice response menus (IVR), special applications, and more.

To access the inbound calls routing table, click the **Incoming Calls** menu item. The following should appear on your screen:

Incoming Call Rules		
S.No	Incoming Rule	Options
1	Route all unmatched incoming calls from provider 'Ports 1,2,3' to '6000 -- Nir Simionovich'	Edit Delete

The incoming rule appears in a human-readable form similar to the syntax used by the outbound routing table.

Click the **Edit** link, located on the right side of the rule. The following dialog should appear on your screen:

Route

from provider

to extension

The above window describes an inbound route from a group of physical ports, indicated by **Ports 1,2,3**—recall these describe your analog FXO ports. Currently, leave the rule as it is, we will make changes later.

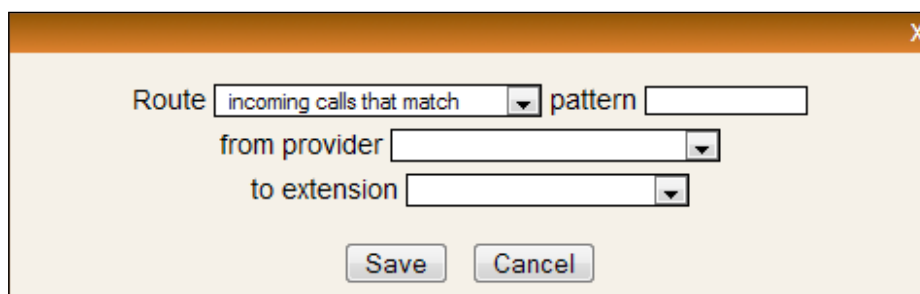
Now, define a new DID-based inbound route. Click the **Add a Incoming Rule** (yes, there is a grammar mistake in the web interface, after all, it is a beta version). The following dialog should appear on your screen:

Route

from provider

to extension

Change the **Route** option from **All Unmatched incoming calls** to **incoming calls that match**. At this point, the dialog box will change to the following:



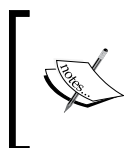
Route pattern

from provider

to extension

At this point, you can choose the service provider link the call will be originating from (**from provider**) and the target to send the call to (**to extension**). However, the question remains, "What should you configure in the pattern text box?" – Well, examine this.

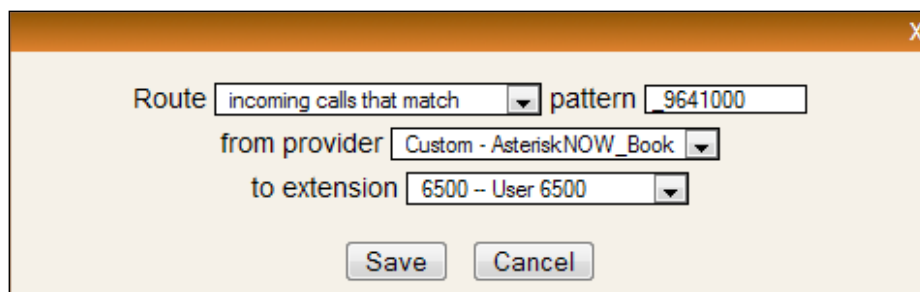
As you may recall from the previous chapter, Asterisk and AsteriskNOW are capable of performing pattern matching according to a specially formulated string (if you don't remember, refer to Chapter 5 for more information). The same logic and syntax that was applied before can also be used in this case.



When configuring your PBX for the first time, it would be a good practice to have a "catch-all" rule. A "catch-all" rule will direct all inbound calls to a sink extension, indicating that the rule is misconfigured and still requires some work.

Example 1: Routing in a Single DID Number

To identify a call made to a single DID, simply fill the pattern text box with the target DID number (in accordance with the service providers syntax). This may look like the following:



Route pattern

from provider

to extension

Note the underscore marking (`_`) at the beginning of the pattern indicates the start of a pattern match—if you don't enter it, the AsteriskNOW GUI will automatically add it for you. While this may seem redundant, the marking is highly important to indicate the pattern match. For additional information about how pattern matching works, please refer back to Chapter 5, to the *Manually Editing Dial-Plan Logic* section.

Example 2: Routing in a Range of DID Numbers

When using a digital connection, such as a PRI connection, the service provider may assign a range of DID numbers to your PRI circuit. When a range of DID numbers is assigned to your connection, the range is usually prefixed with a few digits, then a certain range of numbers are assigned as suffix. For example, you may have assigned the numbers 9641000 to 9641029, which indicates that you have assigned 30 DID numbers.

Following are a few examples of pattern strings, matching various DID ranges:

DID Range	Total DID Numbers	Pattern
9641000 - 9641099	100 Numbers	<code>_96410XX</code>
9641000 - 9641009	10 Numbers	<code>_964100X</code>
9641000 - 9641029	30 Numbers	<code>_96410[012]X</code>

Once an inbound call has been identified to match a specific pattern, the call will be directed to the destination indicated by the **to extension** field.



In most realistic scenarios, routing multiple DID numbers into a single location is utilized as a "catch-all" rule, while specific DID numbers are directed at specific extensions or IVR sections—as described in the next chapter.

Inbound Call Routing in `extensions.conf`



CAUTION!

The following section is for advanced users only! If you are not familiar with programming languages, at this point, please skip to the next chapter. Manually editing your AsteriskNOW dial-plan files isn't recommended.

Inbound call routing is performed in the same manner as outbound routing via the `extensions.conf` configuration file. In order to understand how it works, you must first go back to the service providers section. Select the **Advanced** properties of **Custom Voip** service provider.

S.No	Service Provider	Type	
1	Custom - AsteriskNOW_Book	Custom Voip	Options ▼
2	Ports 1,2,3	Analog	Edit
			Codecs
			Advanced
			Delete

The following dialog box should appear on your screen:

Advanced Settings

X

trunkname:

trunk_2

insecure:

port:

4569

Caller Id:

fromdomain:

fromuser:

contact:

update

Cancel

Note the **trunkname** item, which is now indicated by the text **trunk_2**—please note that this may vary on your system. Now, go back to the `extensions.conf` file, and seek the proper location of the inbound routing associated with **trunk_2**.

Inbound routing from services providers is always prefixed with the `DID_` prefix, followed by the trunkname setting. Look for the **DID_trunk_2** section of the `extensions.conf` file.

[DID_trunk_2]

include=default

exten=_9641000,1,Goto(default|6500|1)

Save

Cancel

include=default

exten=_9641000,1,Goto(default|6500|1)

At this point, you may modify the dial plan by adding new pattern matches to it, modify the existing one to perform an additional function before performing its GUI assigned operation, and more.

Summary

In this chapter you learned how calls are routed into your AsteriskNOW PBX system, and how to perform initial handling of these calls. In the next chapters, you will learn about Interactive Voice Response (IVR), voicemail systems, call queues, and more—the building blocks of the modern PBX system.

7

"For Annoyance, Press 1"— Voice Menus and IVR

If you're going through hell, keep going. — Winston Churchill

Interactive Voice Response (IVR)-enabled telephony platforms are the pivot elements of the telephony industry. You must be familiar with several IVR telephony platform—your cellular provider's top-up system, your bank's call distribution system, or your Internet service provider's technical support line—IVR systems are everywhere. As you may already know, IVR systems can be a hell to use, but as Winston Churchill said, you just have to keep on going to get to your destination.

In this chapter you will learn about the human dynamics of IVR systems, the implementation and the deployment of an IVR environment to your AsteriskNOW PBX system.

Four Rules of IVR

IVR systems can be hell to use; the main reason for this is that people designing IVR systems tend to complicate their functionality beyond the normal usage scope of a human being. The following four rules will enable you to implement a usable, humanly accessible, and maintainable IVR.

- **Rule 1—Keep it narrow:** Your IVR should be kept as simple as possible. Make sure that each of your steps in the IVR environment is not longer than five options. Most IVR users aren't able to remember all the options of an IVR system, when presented with a multitude of IVR options.
- **Rule 2—Keep it shallow:** Your IVR system's depth is in direct relation to the complexity of the IVR application. If your application has more than four levels, you need to revise your IVR plan. Most people (including yours truly) become extremely aggravated when confronted with an IVR system that is asking for too much information.

- **Rule 3 – Enable escape routes:** Always give your IVR user the ability to break out of the IVR flow and talk to a live person. Some people simply can't handle the usage of an IVR system.
- **Rule 4 – If it works, don't fix it!:** For some reason, companies tend to change their IVR flows every other day to support a new business model. An IVR system that constantly changes is a nightmare for users, as they never get used to the options of the system. If you must perform an update, make sure that your update doesn't affect that system in such a way that the users need to re-learn the system.

Voice Menus—AsteriskNOW's IVR Generator

AsteriskNOW provides a highly simplistic IVR generator, rightfully named— **Voice Menus**. The main usage of an IVR in a PBX is the implementation of an "Auto Attendant".



Some PBX systems refer to auto-attendant and IVR as two different things. In AsteriskNOW, they are one and the same.

At this point, click the **Voice Menus** link, located on your left-hand main menu. The following should appear on your screen:

The screenshot shows the 'Voice Menus' configuration window. On the left, a list shows 'VoiceMenu - mainmenu' selected. The main area contains fields for 'Name' (mainmenu) and 'Extension'. Below these are 'Steps' with 'Up' and 'Down' buttons, and a list of steps: 'Answer the Call', 'Play \'thank-you-for-calling\' & Listen for KeyPress', 'Play \'If unknown ext dial\' & Listen for KeyPress' (highlighted), 'Play \'otherwise\' & Listen for KeyPress', and 'Play \'pls hold-while try\' & Listen for KeyPress'. There are 'Add' and 'Delete' buttons for steps. A checkbox 'Dial other Extensions?' is checked. Below is a 'Keypress' Events table with columns 'Key' and 'Action'. The table has 7 rows, all with 'Disabled' in the Action column. At the bottom are 'New', 'Delete', 'Save', and 'Cancel' buttons.

Key	Action
0	Disabled
1	Disabled
2	Disabled
3	Disabled
4	Disabled
5	Disabled
6	Disabled

This interface enables editing, creation, or deletion of voice menus. Each menu is built from a set of operational directives (**Steps**) and functional keys (**Keypress Events**). Each voice menu also receives a mandatory name (**Name**), a form of logical entity description, and an **Extension** number (optional). The extension number enables PBX extensions or external users to dial into the specific voice menu indicated by the extension number.

Voice Menu Steps—The Voice Menu Flow

Steps are performed one after the other, in the order they appear on the screen. There are seventeen possible steps, available through the AsteriskNOW GUI.



At the time of writing this book, the number was seventeen. However, it may happen that once published, the number of possible steps will increase.

Once a step had been selected, the GUI will change, indicating the requirement for additional fields to be filled. The seventeen available steps are as follows:

Step Type	Description	Parameters
Answer	Answers the call—usually the first step of every voice menu.	
Authenticate	Asks the user to enter a password using the keypad. The password is compared with the information in the step directive.	Password to authenticate against—numeric only.
Background	Plays a voice file, allowing the user to interrupt the playback by pressing a key.	File to play back from the AsteriskNOW recordings database.
Busy Tone	Generates a busy tone to the calling user.	
Congestion	Indicates congestion to the calling user.	
DigitTimeout	Sets the maximum timeout between digit presses.	Time in seconds.
DISA	Activates the DISA application (DISA—Direct Inward System Access).	DISA authentication password—numeric only.
ResponseTimeout	Sets the maximum timeout waiting for a response from the user.	Time in seconds.

Step Type	Description	Parameters
Playback	Plays a voice file without interruptions from the user.	File to play back from the AsteriskNOW recordings database.
Wait	Waits a certain amount of time.	Time in seconds.
WaitExten	Waits a certain amount of time for the user to dial a new extension number.	Time in seconds.
Goto Menu	Goes to a predefined voice menu.	Voice menu name from the database.
Goto Directory	Goes to the PBX directory system.	
Goto Extension	Rings a specific extension.	Extension number from the AsteriskNOW users database.
Goto TimeBasedRule	Goes to a predefined time based rule.	Time Based Rule from the AsteriskNOW time rule database.
Dial RingGroup	Rings a predefined number of extensions.	Ring group number from the AsteriskNOW ring group database.
Hangup	Hangs up the call.	

At this point, you may have encountered new elements, about which you have no knowledge of what they are and how they work. These include the following: DISA, Recordings, Time Based Rules and Ring Groups. Let's learn more about the step types – what they are, how they are configured in AsteriskNOW, and later, how they are used within the Voice Menu construct.

DISA—Direct Inward System Access

DISA operation enables an enterprise to allow its users to dial in to the enterprise PBX system, authenticate via a personal code, and then utilize the enterprise PBX system to perform dialling functions, as if they were located back at their office desk.

Most installations utilize DISA services to allow their employees access to long distance calling facilities, without enabling these calling facilities from their cellular phones. DISA services became especially popular with the introduction of cellular VPN (Virtual Private Network) services, creating a zero-charge group between an enterprise's PBX system and an employee's cellular phone.

Recordings—Menus and System Playbacks

Like any normal PBX, AsteriskNOW requires recorded messages to provide the various menus and user messages building up the user interaction. Now, while the following may sound a little confusing, message recording is performed utilizing the **Record a Menu** link, from the main menu — don't let the name of the link fool you — once a file is recorded, it can be utilized as either a menu or a regular announcement.

The process of recording a new announcement or voice menu works as follows:

Step 1: Where am I?

To record a new voice menu or announcement, you need one of the PBX extensions next to you. The phone's handset is used as the means of recording your voice menu or announcement.

Assume that you are currently located next to extension number 6501.

Step 2: Prepare Your Text

For a small office PBX the text played back isn't crucial, however, for enterprises the spoken text can be crucial. If you want to have a well formulated text, use the services of a copy writer. Write your announcement on a piece of paper, and when recording simply read it from the paper, don't think about what you're saying while recording — it will simply sound bad.

For the purpose of our task, record the following messages:

Message name: AsteriskNOW_Announce

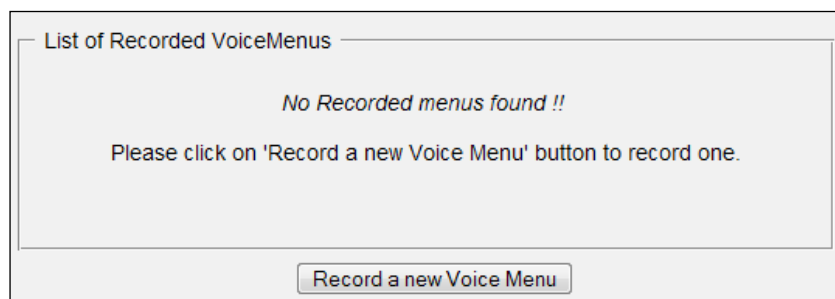
"Congratulations — you have successfully installed and executed the AsteriskNOW PBX system."

Message name: AsteriskNOW_Select

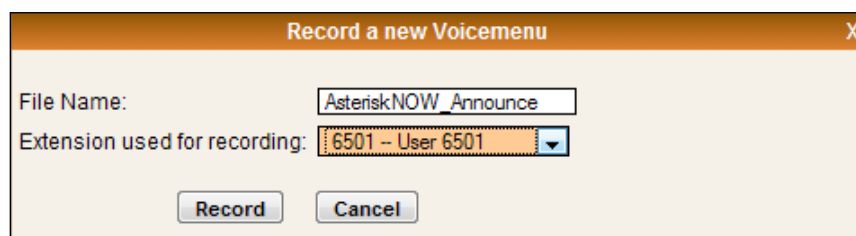
"You may now dial the user extension number you wish to dial."

Step 3: Recording

Now you will perform the actual recording of the audio file. First, click the **Record a Menu** link, located in the main menu. The following should appear on your screen:



Click the **Record a new Voice Menu** button. The following dialog should appear on your screen:



The above information doesn't appear on your screen, so fill out the relevant information in the dialog box. Once you have entered the information, pressing the **Record** button will ring the extension number indicated – in your case, extension number **6501**. A prompt indicating you to start recording at the sound of the beep will be heard (this is a fairly fast prompt, so wait for the beep). Once you have finished your recording, simply press the # button on your keypad, or hang up.



BUG Note:

As indicated before, the version of AsteriskNOW used for the writing of this book is Beta-6. This version contains a small bug in regards to refreshing the list of recorded messages. Once you have recorded your message, do not forget to switch to another section, and then back to **Record a Menu**, to refresh the interface—hitting the refresh button on your browser will refresh everything.

Once you have refreshed your screen, the following should appear on your screen:

List of Recorded VoiceMenus			
S.No	Name	Options	
1	AsteriskNOW_Announce.gsm	Record Again	Play Delete

At this point, you can re-record your message, using the **Record Again** button, **Play** the recording, or **Delete** it. Recording/Playing is performed via a connected phone, so make sure you have a phone next to you.

Time Based Rules

Time based rules are a means of letting a certain portion of the PBX behave in some way during specific time frames and another way in others. A good thing about time based rules is that they can be cascaded to perform a very complicated time selection scenario.

In order to create a new time based rule, click the **Time Based Rules** option; the following screen should appear:

Time Based Rules 

No Previous Time Rules found !!

Please click on the 'New Time Rule' button
to define a New Time Based Rule

When defining a new time based rule, the following information should be taken into consideration (all fields below are mandatory):

Add new Time Rule X

Rule Name : (Ex: July4)

Time & Date Conditions

Start Time: : End Time: :

Start Day: End Day:

Start Date: End Date:

Start Month: End Month:

Destination

if time matches:

if time did not match:

Field Name	Description	Applicable Values
Rule Name	A descriptive name for your time based rule.	Any string of characters.
Start Time & End Time	The time period in the day the rule applies in.	Time selection by hours and minutes from the selection box.
Start Day & End Day	The days of the week the rule applies in.	Consecutive days period of the week.
Start Date & End Date	The period of the month the rule applies in.	Consecutive dates period of the month.
Start Month & End Month	The months of the year the rule applies in.	Consecutive months period of the year.
Destination if time matches	A destination within the PBX to which to route the call, if the time period matches.	An already configured resource of the PBX.
Destination if time did not match	A destination within the PBX to which to route the call, if the time period doesn't match.	An already configured resource of the PBX.

**IMPORTANT:**

If you wish to configure cascading time based rules, upon completing each rule, you **must** click the **Activate Changes** button, so that the last created rule becomes available in the selection box.

For the purpose of configuring your PBX create a time based rule to indicate the 4th of July. Create a new time based rule, called **July 4th** and fill in the information provided below:

Add new Time Rule [X]

Rule Name : (Ex: July4)

Time & Date Conditions

Start Time: : End Time: :

Start Day: End Day:

Start Date: End Date:

Start Month: End Month:

Destination

if time matches:

if time did not match:

Ring Groups

Ring groups are sometimes referred to as "hunt-groups" (especially in Europe). The purpose of a ring group is to provide a facility in which multiple connected handsets can ring at the same time. Ring groups are not limited by the technology of the connected handset, meaning that a ring group can contain a mixture of analog and VoIP handsets at the same time.

To create a new ring group, click the **Ring Groups** option; the following screen should appear:

Ring Groups [New Ring Group]

No Previous Ring Groups found !!

To create a ring group click on the 'New Ring Group' button

Now, click the **New Ring Group** button, in order to open the **Add Ring Group** dialog box:

Add Ring Group

Name: Strategy:

Ring Group Members

- SIP/6500
- SIP/6501
- IAX2/6500

Available Channels

- Zap/4
- SIP/6000

Extension for this ring group (optional) :

Ring (each/all) for these many seconds :

If not answered

☒ Goto Voicemail of this user ☐ Goto an IVR menu ☐ HangUp

As you can see in the above screenshot, the dialog box contains two lists and some configurable options. The list on the left indicates the members of the configured ring group, while the list on the right indicates the available extensions that are, currently not part of the current ring group. The options available are as follows:

Field Name	Description	Applicable Values
Name (Mandatory)	Descriptive name for your ring group.	Any string of characters.
Strategy (Mandatory)	The methodology with which the extensions ring in the ring group.	Ring In Order – Ring the extensions one after the other. Ring All – Ring all configured extensions together.
Extension for this ring group (optional)	A virtual extension; when dialled it will ring the configured ring group.	A numeric value.
Ring (each/all) for these many seconds (Mandatory)	Ring each extension on the ring group, or the entire ring group (depending on the strategy) for this many seconds.	A numeric value. Recommended values are between 15 and 20 seconds.

If a ring group is rung and no extension picks up the call, the call is flagged as "not-answered", immediately failing into the "If not answered" rule. If a call is not answered, the call falls back to one of the following three options:

- **Goto Voicemail of this user:** Will automatically direct the un-answered call to the voicemail indicated by the selection box.
- **Goto an IVR menu:** Will automatically direct the un-answered call to a pre-configured voice menu.
- **HangUp:** Simply hang up the un-answered call.

Enough Theory, Back to Voice Menus

Last time, your voice menu section included a single voice menu, designated as mainmenu. First of all, assign a PBX extension to your main menu, so that you can listen to it.

Click the **Voice Menus** menu item and then select the **Voicemenu-mainmenu** from the selection box on the left side of the screen. The form on the right should now be filled with the various steps of the mainmenu voice menu. The following should appear on your screen:

Voice Menu Configuration

Voice Menus:

VoiceMenu - mainmenu

Name: mainmenu Extension:

Steps:

Up Down

Answer the Call
 Play 'thank-you-for-calling' & Listen for KeyPress
 Play 'if-u-know-ext-dial' & Listen for KeyPress
 Play 'otherwise' & Listen for KeyPress
 Play 'pls-hold-while-try' & Listen for KeyPress

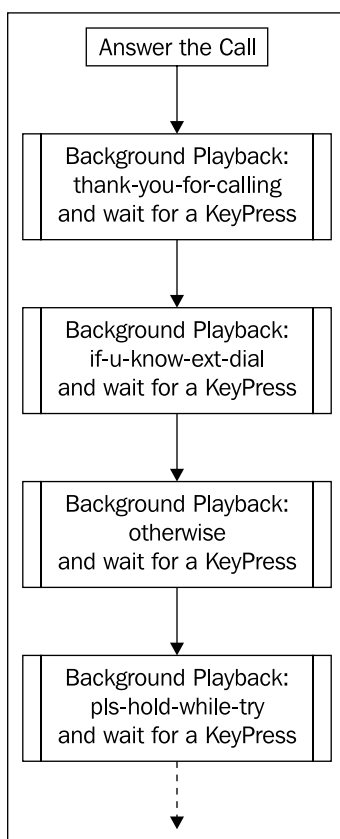
Add a new Step:

- Select - Add Delete

☒ Dial other Extensions?

Enter the number **7501** in the text box designated **Extension** then click the **Save** button. Activate your changes (as always) and try dialling 7501 from a phone connected to the PBX. If all goes well, you should be listening to the main menu. Pay attention to the announcement you listen to and the contents of the **Steps** list.

As you may have noticed, each of the steps corresponds to a specific voice announcement you have heard on your phone. AsteriskNOW simply goes through the steps list, one by one, and performs the steps as they are indicated. If we were to translate the current mainmenu voice menu into a flow chart, it would look something like the flow chart below.



The flowchart is not a complete one, it is a partial look into the mainmenu pre-configured voice menu.

As you may have noticed, the mainmenu currently configured isn't very useful to us, as it doesn't really wait for information to be entered, and if no information is entered, it simply hangs up. You shall now modify the mainmenu to a voice menu that will serve your PBX better.

The behavior of your main menu as a simple step-based list is as follows:

Step 1: Answer the call.

Step 2: Play the "thank-you-for-calling" recording, allowing playback to be interrupted by input.

Step 3: Play the "if-u-know-ext-dial" recording, allowing playback to be interrupted by input.

Step 4: Play the "otherwise" recording, allowing playback to be interrupted by input.

Step 5: Play the "please-hold-while-try" recording, allowing playback to be interrupted by input.

Step 6: Wait for an extension number to be entered for 5 seconds.

If no extension had been entered during the course of your voice menu, ring the operator, located at extension 6500.

Most of the above is already configured, but you are most probably wondering how to transfer the call to extension 6500, if no extension has been selected and if the 5 seconds has timed out.

To configure the above:

Step 1: Remove the last step from the pre-configured steps list (this step is not required).

Step 2: Add the "WaitExten" step, indicating a 5 seconds delay.

Step 3: Scroll down the "Keypress Events" select box and locate the "t" event" – indicating a timeout has occurred.

Step 4: Modify the event from "Disabled" to "Goto Extension" and select the extension to be 6500.

Step 5: Modify the event of the "i event" to be identical to that of the "t event". The "i event" indicates that an invalid extension number was entered.

At this point, your **VoiceMenu mainmenu** should look as follows:

The screenshot shows a configuration window titled "Voice Menu:". On the left, a list box contains "VoiceMenu - mainmenu". To the right, the "Name" field is "mainmenu" and the "Extension" field is "7501". Below these, the "Steps" section contains a list of five steps: "Play 'thank-you-for-calling' & Listen for KeyPress", "Play 'if-u-know-ext-dial' & Listen for KeyPress", "Play 'otherwise' & Listen for KeyPress", "Play 'pls-hold-while-try' & Listen for KeyPress", and "Wait Exten '5' sec". There are "Up" and "Down" buttons for the steps. Below the steps, there is a section "Add a new Step:" with a dropdown menu showing "-- Select --", and "Add" and "Delete" buttons. A checkbox labeled "Dial other Extensions?" is unchecked. Below that is a section titled "'Keypress' Events" which contains a table with two columns: "Key" and "Action".

Key	Action
7	Disabled
8	Disabled
9	Disabled
*	Disabled
#	Disabled
t	Goto Extension 6500
i	Goto Extension 6000

At the bottom of the window are buttons for "New", "Delete", "Save", and "Cancel".

After modifying the configuration, click the **Save** button and then activate the changes. Now, pick up your phone and dial the 7501 extension, if all goes well, you should hear the announcements, and then if you haven't pressed anything, you should be transferred to extension 6501.

Playback and Background Steps



Playback and background require a recorded file name to work. As you may have noticed, if you add one of these steps, you will be asked to fill a text box. If you click on the text box, a small window will pop up. This is a scroll window, although it may not look like it (browser issues with some browser versions), simply use the arrow keys to scroll up and down to locate your recording.

Summary

You have just mastered one of the basic elements of modern PBX systems. A proper combination of voice menus and carefully planned recordings will provide a very enjoyable IVR experience for your PBX users, making sure that they don't get aggravated by your PBX system. Try to avoid the concept of making your company look bigger than it really is, by complicating your voice menus — keep it simple; don't add a menu when it's not required.

8

Voicemail, Conferencing, and Parking—Advanced PBX Services

I have had fans make me the big picture collages of the photo books; I have had fans send me birthday cakes... sing to me on my voicemail. I have had fans flash me. I have had older fans give me their bras and underwear onstage. — P. Diddy

While Puff Diddy's fans appear to make a very artistic usage of his voicemail, voicemail systems and conferencing systems provide a driving force in today's modern business. The abilities to accept your messages when you're out of the office, get them delivered to you while you're on the move, and hold telephone conferences from any location in the world are radically changing the way businesses work today. Even a small operation can enjoy these benefits and gradually allow for more elaborate communications with its customers. AsteriskNOW provides built-in facilities for both voicemail and conferencing services.

Comedian Mail—The Asterisk Voicemail System



Consider the following statement: "There is nothing comic or funny about the Asterisk voicemail system!" — So why is it called "Comedian Mail"? According to the Asterisk community folklore, the name "Comedian Mail" was derived from Nortel's "Merdian Mail". Essentially, the user interaction of Asterisk's voicemail system is identical to that of Nortel's Meridian, thus, the name was metamorphosed into Comedian.

As you may recall, in Chapter 3, you have already provisioned your user extensions with voice mailboxes. So, essentially speaking, there should be no other configuration in regards to the extensions' mailboxes – and that is true. AsteriskNOW provides a facility to define the general aspects of how the voicemail system operates across the entire PBX system. This facility is available via the **Voicemail** link – from the main menu. Once you click the **Voicemail** link, the following screen should appear:

Voicemail Settings:

6000 -- Nir Simionovich
6500 -- User 6500
6501 -- User 6501
7501 -- Voice Menu
8500 -- Check Voicemail
8700 -- Fax

Extension for checking messages: 8500

Attach recordings to e-mail: ☒

Max greeting (seconds): 60

Dial '0' for Operator: ☐

Message Options:

Message Format:

Maximum messages : 100 (per folder)

Max message time: 2 minutes

Min message time:

Playback Options:

☐ Send messages by e-mail only

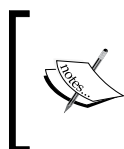
☐ Say message Caller-ID

☐ Say message duration

☐ Play envelope

☐ Allow users to review

Save Cancel



As you may notice, some of the entries on the left list box appear to be grayed out. The reason for this is that these items are not controllable from this configuration area. If you were to select one of these, you would be greeted with an alert explaining where that specific entry can be edited.

Each of the configuration options available for the voicemail system is explained in the following sections.

Voicemail General Options

General options are used to define global functionalities of the voicemail system, not the way messages are recorded or handled.

Field Name	Description	Applicable Values
Extension for checking messages	The extension number a user needs to dial in order to check their voicemail.	Any numerical extension
Attach recordings to e-mail	Do you want to send the voicemail system recording to the user's mailbox via email?	On or Off
Max greeting (seconds)	What is the limit of the users personal greeting, in seconds?	Any numerical value
Dial '0' for operator	Should you allow the user to dial "0" from the voicemail system to break back into the previous menu or reach an operator?	On or Off

Voicemail Message Options

The following options relate directly to how messages are stored and managed for each of the mailboxes in the PBX.

Field Name	Description	Applicable Values
Message format	What format should be used to save the voicemail messages? If you intend to send the voicemail by e-mail, use one of the WAV formats.	WAV (GSM) WAV (16 bit) Raw GSM
Maximum Messages	Maximum number of messages per voicemail folder.	10 25 100 – default 250 500 1000
Max message time	The maximum duration for a voicemail message.	1 minute 2 minutes – default 5 minutes 15 minutes 30 minutes unlimited
Min message time	The minimum time for recording a message; any message recorded will be at least of that length.	1 second 2 seconds 3 seconds 4 seconds 5 seconds No minimum – default



The format chosen has a direct impact on the amount of space required on the hard drive for the storage of each message. An optimal setting for a system that would send messages via e-mail would be to use WAV (with the voice encoded as GSM).

Voicemail Playback Options

Field Name	Description	Applicable Values
Send messages by e-mail only	Sends the voicemail message to the mailbox's designated e-mail, without saving the message on the system.	On or Off
Say message Caller-ID	Announces the caller ID of the caller (if available) to the mailbox owner, when retrieving messages.	On or Off
Say message duration	Announces the message duration, prior to playback.	On or Off
Play envelope	Plays the voicemail instructions.	On or Off
Allow users to review	Allows users to review their messages before actually saving them to the voicemail system.	On or Off

MeetMe Conferencing



MeetMe conferencing should not be confused with three-way conferencing. The conferencing facility AsteriskNOW provides is a multiple, dial-in, conferencing facility.

Conferencing (or tele-conferencing as some may call it) provides one of the most essential tools for the modern business—a way to converse with multiple people at the same time. A conference call is activated according to the following flow:

- **Step 1:** The users will dial in to the designated conference room extension.
- **Step 2:** The users will authenticate themselves to the system, in accordance with their designated passwords.
- **Step 3:** The users will start conversing using the conference bridge, in accordance with the conference bridge initial configuration.

This flow is common among most, if not all, the conferencing systems around the world. Each step of the conference bridge is clearly announced, allowing the user to easily traverse the system.

Conference User and Administrator Key Presses

Following is a summary of the various key-press operations available for a conference user and the conference administrator. These key presses can be used while participating in a conference, enabling some control over the conference experience.

Key Press	User Function	Administrative Function
1	Mute or un-mute yourself within the conference.	Mute or un-mute yourself within the conference.
2	N/A	Lock or unlock the conference — doesn't allow new users to enter the conference.
3	N/A	Eject the last user that entered the conference.
4	Decrease conference volume.	Decrease conference volume.
5	N/A	N/A
6	Increase conference volume.	Increase conference volume.
7	Decrease your own volume.	Decrease your own volume.
8	Exit the conference.	Exit the conference.
9	Increase your own volume.	Increase your own volume.
*	Play back the administrative menu.	Play back the administrative menu.

Defining a New Conference Room

To define a new conference room, click the **Conferencing** link on the main menu; the following will appear on your screen:

Conference Bridges:

- 6000 -- Nir Simionovich
- 6500 -- User 6500
- 6501 -- User 6501
- 7501 -- Voice Menu
- 8500 -- Check Voicemail
- 8700 -- Fax
- New Entry**

Extension:

Room Override:

Password Settings:

PIN Code:

Admin PIN Code:

Conference Room Options:

- ☒ Play hold music for first caller
- ☐ Enable caller menu
- ☒ Announce callers
- ☐ Record conference
- ☐ Quiet Mode
- ☐ Wait for marked user
- ☐ Set marked user

General Conference Options

General options are used to define global settings of the conference room, not the way users interact with the conference room.

Field Name	Description	Applicable Values
Extension	The extension number to be used for the conference room. The extension number also designates the room number.	Any numerical value.
Room Override	Room override provides a facility to create aliases for the same conference room, but with different room options. The password fields are not applicable for conference room aliases.	An already existing conference room extension number.

Conference Password Settings

Passwords are required to identify between the conference administrator and a regular conference user.

Field Name	Description	Applicable Values
PIN Code	A numerical password identifying a regular user with the conferencing system.	Any numerical value.
Admin PIN code	A numerical password identifying an administrative user with the conferencing system.	Any numerical value.

The numerical code will distinguish between a normal conference user and the conference administrator. The administrator is able to perform some global conference functions – locking the conference and removing the last user that joined.

Conference Room Options

The conference room options control the way the conference room interacts with the users.

Option	Description
Play hold music for first caller	Plays on hold music to the user when the first user enters the conference and no one else is yet in the conference.
Enable caller menu	Allows the user to press the * key to gain access to the user menu.
Announce callers	Asks users to record their name and announce their arrival to the conference room, prior to inserting them into the conference room.
Record conference	Records the conference to a WAV file. The file will be located in the recordings directory, named according to the following format: Filename: meetme-conf-rec- <code>\${conference_number}</code> - <code>\${uniqueid}</code>
Quiet Mode	If the conference is set to quiet mode, all users entering the conference will be in Listen-Only mode (globally muted).
Wait for marked user	If this option is set, users currently in the conference will not be able to talk to one another, till the marked user enters the conference room.
Set marked user	The user that has called this conference extension is set as the marked user – enabling the usage of the previous option.

It is imperative to understand the relation between conference room aliasing and marked users. Essentially, when you wish to create a conference room that will not allow users to talk until the administrative user enters the conference, configure according to the following:

- **Step 1:** Create the initial conference room, setting the room with **Wait for marked user** and proper PIN codes.
- **Step 2:** Create a new extension number with a **Room Override** pointing to the previously created conference room. Set the new extension number with the **Set marked user** option.

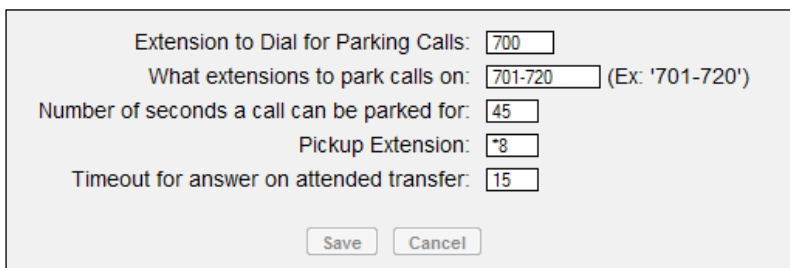
Now, your users will call the main conference room, while the administrator will call the aliased conference room. Once the administrator is in the room, the participants will be able to talk.

Call Parking

Call parking provides a facility for placing calls into a specific area of the PBX and then retrieving them. The concept of operation is to answer a call, park the call on an extension, and pick up the call from another extension or from the same one.

One of the most common uses for call parking is consultation calls. Consultation calls are situations where a call has been answered by an extension. The user of that extension needs to consult with someone else. The user will transfer the current call to the parking lot (basically transferring the call to the parking lot extension), get a parking lot position, call a new extension, consult with another user on the PBX, then retrieve the previous call from the parking lot and continue it.

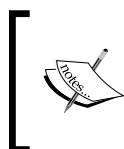
In order to define call-parking features, click the **Call Parking** link on the main menu; the following screen will be obtained:



A configuration dialog box for call parking with the following fields and buttons:

- Extension to Dial for Parking Calls:
- What extensions to park calls on: (Ex: '701-720')
- Number of seconds a call can be parked for:
- Pickup Extension:
- Timeout for answer on attended transfer:
- Buttons:

Option	Description
Extension to Dial for Parking Calls	The extension identifying the PBX parking lot. Once a call has been parked in the parking lot, an announcement will indicate the number of the parking lot.
What extensions to park call on	The size of the parking lot; each number in the range designates a parking lot position. The number announced can be dialed from any extension connected, to re-connect to the previously parked call.
Number of seconds a call can be parked	The duration (in seconds) for which calls can be parked in the parking lot
Pickup Extension	Extension to dial to pickup a dialing extension.
Timeout for answer on attended transfer	When performing an attended transfer, how long to wait for an answer from the called party.



Additional call features can be configured manually via the `features.conf` file. Information about the `features.conf` file can be obtained at: <http://www.voip-info.org/wiki/index.php?page=Asterisk+config+features.conf>

Summary

Conferencing, voicemail, and call parking are some of the most useful features of the modern PBX system. Unlike most PBX systems, which require additional hardware to facilitate the above services, Asterisk and AsteriskNOW provide these features as built-in functions, thus lowering the cost of maintaining similar services in the enterprise.

9

"Please hold, we'll be with you shortly"—Simple Call Queues

An Englishman, even if he is alone, forms an orderly queue of one.
— George Mikes

I would like to make the following clear—I love England! I really do, London, to me, is one of the most amazing places in the world and there is something about England that is intoxicating. Nonetheless, Englishmen are so well behaved that, even when they are the first in line, they will form their own queue.

Queues are usually associated with call-center operations, and usually, people hate phone queues. There is nothing more annoying than being put into a queue, listening to the most annoying music in the world, and once in a while hearing a message of the form *your call is important to us, please continue to hold the line....* AsteriskNOW provides a facility to build call queues. The purpose of this chapter is to get you acquainted with AsteriskNOW call queues, how to configure them and control them, and their utilization with AsteriskNOW.

To define a new call queue, click the **Call Queues** link on the main menu; the following will appear on your screen:

Queue General Options

General options are used to define global settings of the call queue.

Field Name	Description	Applicable Values
Queue	The queue's extension number.	Any numerical value.
Full name	A descriptive name for your queue.	Any string of characters.
Strategy	The methodology with which the queue's agents are rung.	RingAll – Ring all available agents till one answers. RoundRobin – Take turns ringing each available agent. LeastRecent – Ring the agent last was least recently called. FewestCalls – Ring the agent with the fewest completed calls. Random – Ring a random agent. RRmemory – RoundRobin with memory. Remembers where it left off in the last ring pass.
Agents	The agents assigned to the queue.	Set each of the agents to be assigned to the queue.

Queue Options

Queue options control the aspects of how calls currently in the queue are handled.

Field Name	Description	Applicable Values
TimeOut	How long to ring an agents extension before passing the call to the next agent	Time in seconds.
Wrapup Time	Upon completion of a successful call to an agent, how much time the agent must remain free, before receiving another call.	Time in seconds.
Max Len	Set the maximum number of callers in the queue.	Any numerical value. 0 means unlimited.
Music On Hold	The music on hold class to play to the callers when they're waiting in the queue.	Selection from a predefined set.
Auto Fill	Should calls be routed to multiple agents simultaneously or wait for each call to be completed?	Yes or No.
Auto Pause	Should calls to failed agents automatically pause the agent so they will not receive additional calls?	Yes or No.
JoinEmpty	Allow callers to join the queue, even if there are no agents in the queue.	Yes or No.
LeaveWhenEmpty	Exit all callers from the queue if there are no agents in the queue.	Yes or No.
Report Hold Time	Should the agent get a report of how long the caller has been waiting in the queue, prior to connecting the caller to the agent?	Yes or No.

Utilizing Call Queues

Call queues, like ring groups, can be utilized either from voice menus or directly from inbound routes ("Incoming Calls") routing tables. It is a fairly common practice to create multiple queues to define multiple skills within a single PBX environment.

For example, imagine that you have a small customer care department. Most agents speak English; some speak Hindi, some Chinese, and some Hebrew. For a call to reach its appropriate agent, you will create four different queues for the customer care department, each representing a different language. Each queue will be assigned with the corresponding agents; thus, you can surely say that if we call into the Hindi queue, the agent that will answer us will talk Hindi.

In accordance with this scenario, your queue configuration will take you through the following steps:

Step 1: Define the Extensions for Each Agent

Assume that your extensions and agents are configured according to the following table:

<i>Extension</i>	<i>Hebrew</i>	<i>English</i>	<i>Hindi</i>	<i>Chinese</i>
200	Yes	Yes		
201	Yes		Yes	
202		Yes	Yes	Yes
203		Yes		Yes
204	Yes			
205		Yes		
206			Yes	

Step 2: Define the Queues

Assume that you want to create a queue for each of the languages. Your queues would look something like the following table:

<i>Queue</i>	<i>Hebrew</i>	<i>English</i>	<i>Hindi</i>	<i>Chinese</i>
2000	Yes			
2001		Yes		
2002			Yes	
2003				Yes

Step 3: Assign the Agents to the Proper Queues

Now, your agents; configuration for each of the queues would look like the following table:

<i>Queue</i>	<i>Agent 200</i>	<i>Agent 201</i>	<i>Agent 202</i>	<i>Agent 203</i>	<i>Agent 204</i>	<i>Agent 205</i>	<i>Agent 206</i>
Hebrew	Yes	Yes			Yes		
English	Yes		Yes	Yes		Yes	
Hindi		Yes	Yes				Yes
Chinese			Yes	Yes			

Now, what remains is to create an appropriate voice menu, indicating to select the language of choice, and the call will be directed to the most fitting agent. This methodology is commonly referred to as "Skill-Based Routing". While skill-based routing in platforms like Avaya is highly complex, a careful queue plan and IVR plan can achieve a similar result.

Summary

Queues provide a facility to control the flow of incoming calls. It is imperative that queues are used with caution, as misconfigured queue behavior can easily make for a very troubled telephony experience and most importantly, a fairly aggravated customer at the other end of the line.

10

General AsteriskNOW Management—Monitoring, Backups, and More

I have pushed the boat out as far as I should in terms of taking on too many things. I'm getting older and I just could not take it any more. I am now monitoring myself very closely and I'm just trying not to get into that sort of state again.
— Stephen Fry

One of the most crucial elements of system administration and system maintenance is the ongoing monitoring to verify if a system is working well. In addition to monitoring, ongoing maintenance such as backup contributes to the general well-being of a PBX system. This chapter will detail the various mechanisms AsteriskNOW offers in regards to general systems management and monitoring.

AsteriskNOW General Options

AsteriskNOW provides a facility to control the general aspects of your PBX functionality, mainly configurations relating directly to the way your PBX behaves and general users settings. These settings are available via the **Options** link, from the main menu. Click on the **Options** link of your main menu; the following screen should be observed:

Local Extension Settings:

Local Extensions are
First Extension Number :
Operator Extension :
☐ Allow analog phones to be assigned to multiple extensions
☐ Allow extensions to be AlphaNumeric (SIP/IAX users)

Agent Login Settings:

Agent Login Extension :
Agent Callback Login Extension :

Default Settings for a New User:

☐ Is Agent
☐ In Directory
☒ SIP
☒ Call Waiting
☐ Voicemail
☐ CTI
☒ IAX
☒ 3-Way Calling
 VoiceMail Password

Local Extension Settings

Local Extension Settings give a general picture of what your extensions space is, where the operator is located, and a few more options. The following table explains the options available.

Option	Description
Local Extensions are	The length of your local extensions, 2 or more digits.
First Extension Number	The number of the first extension to be created.
Operator Extension	Extension to be used as the general operator extension.
Allow analog phones to be assigned to multiple extensions	Should the PBX allow the creation of multiple extension numbers to be assigned to a single analog port (FXS)?
Allow extensions to be AlphaNumeric (SIP/IAX)	Can VoIP extensions utilize alphanumeric characters? Hybrid PBX systems (TDM + VoIP) shouldn't allow this functionality, in order to allow uniformity.

Agent Login Settings

Agents assigned to queues need to log in to their designated queue to flag to the queue that they are now available to accept calls.

Option	Description
Agent Login Extension	This extension will provide a login point to the agent's designated queue. The username/password combination is the same as that of a user (User Extension/User Password). Once an agent has logged in, he or she will hear Music On Hold. Once the handset has been disconnected, the agent will be automatically logged out.
Agent Callback Login Extension	This extension will provide a login point to the agent's designated queue. The username/password combination is the same as that of a user (User Extension/User Password). Once an agent has logged in, he or she will be prompted for a callback extension, designating where the agent is waiting for calls to be connected to. This way, the agent isn't required to remain on the line to receive new phone calls.



While it may appear that agents are able to log in but not log out of the queue (most probably a bug in Beta 6 of AsteriskNOW), a workaround is to log in via the **Agent Callback Login Extension** and in order to log out, perform a new login via **Agent Login Extension**.

Extension Options

The extension options provide a template for extension creation, when creating new extensions in the user's administration section.

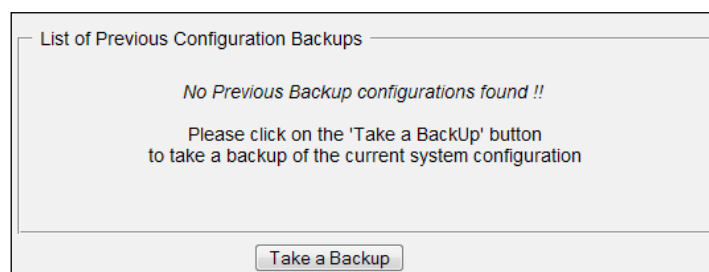
Option	Description
Is Agent	Will this device or user be part of a calls queue?
In Directory	Should our provisioned user be listed in the PBX directory?
SIP	Should our device be a SIP-based device?
Call Waiting	Should our device or user have call waiting enabled?
Voicemail	Should our user have a voicemail mailbox?
CTI	Should our user be defined with Asterisk Manager login abilities?
IAX	Should our device be an IAX device?

Option	Description
3-Way Calling	Is our device capable of rendering 3-way calling features?
Voicemail Password	Default voicemail password.

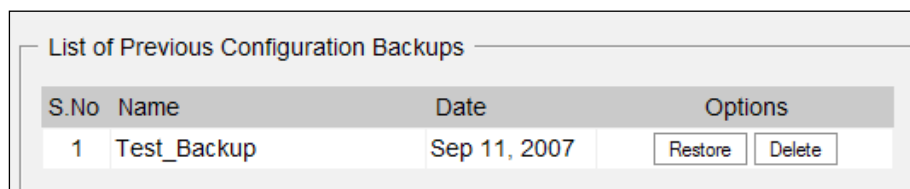
AsteriskNOW Backup

Backups are a crucial part of the well-being of every system well being. A proper configuration backup methodology provides a way to restore from a system disaster, or in some case, a means of restoring your system in a cases of re-installation.

AsteriskNOW provides a fairly simplistic backup facility, via the **Backup** link from the main menu. Click on the **Backup** link of your main menu; the following will screen should be observed:



Once you click the **Take a Backup** button, a backup of your current configuration will be taken. You must specify a backup name, prior to the performance of a backup. Once you have performed your backup, the screen will change, and the following screen should be observed:



List of Previous Configuration Backups			
S.No	Name	Date	Options
1	Test_Backup	Sep 11, 2007	Restore Delete

Currently, this backup mechanism is meant for configuration backup after completing an installation, or after completing a configuration change. This is not intended to be utilized as a daily backup procedure.

One of the nice things about the AsteriskNOW GUI is its dynamic nature. Unlike traditional Asterisk GUIs (such as FreePBX, VoiceOne, and others), AsteriskNOW works directly with the operating system's file system; allowing it to show changes in the file system directly to the GUI, immediately after they are performed. This enables an interesting functionality, when combined with the backup function.

All backups performed by AsteriskNOW are saved as `tar.gz` files in the `/var/lib/asterisk/gui_configbackups` directory. If you were to copy these configuration backups out of the PBX, you would have a complete live working backup of your configuration.



If you wish to back up your AsteriskNOW installation, the best thing to back up would be the `/var/lib/asterisk` directory, as it will contain both backups of configurations and sound files.

If you wish to install a third-party backup agent on your AsteriskNOW system, please refer to Chapter 11 for additional information regarding advanced configurations.

Asterisk Logs

The Asterisk Log messages provide an insight into the Asterisk running console, directly from the web interface. While this is not a progressing log, it is a handy tool for performing first-level debugging of issues.

As a default, the log is configured to show messages categorized as an `ERROR`, a `WARNING`, or a `NOTICE`. The format of the log is as follows:

```
[Sep  9 11:25:52] ERROR[3098] res_config_pgsql.c: Postgresql RealTime:
Failed to connect database server asterisk on 127.0.0.1. Check debug
for more info.
[Sep  9 11:25:52] WARNING[3098] res_config_pgsql.c: Postgresql
RealTime: Couldn't establish connection. Check debug.
[Sep  9 11:25:52] NOTICE[3098] config.c: Registered Config Engine
pgsql
[Sep  9 11:25:52] WARNING[3098] app_queue.c: Unknown keyword in queue
'5000': fullname at line 53 of queues.conf
[Sep  9 11:25:52] NOTICE[3098] app_queue.c: The 'roundrobin' queue
strategy is deprecated. Please use the 'rrmemory' strategy instead.
[Sep  9 11:25:52] ERROR[3098] chan_misdn.c: Unable to initialize mISDN
```

Please note that each line is prefixed with its proper severity level, indicating how to relate to the information provided. Following is a table containing the various severity levels, as indicated in the log file:

Severity Level	Description
DEBUG	Debug-level messages are indicated for programmers who wish to understand inner workings of Asterisk errors when they occur. Debug-level messages can also provide insight when debugging various VoIP connections.
NOTICE	Notice messages indicate information that should be noted, but has no meaningful impact upon Asterisk's functionality.
WARNING	Warning-level messages indicate that something has happened that requires analysis, has some form of impact on Asterisk's functionality, but has not caused a failure.
ERROR	Error-level messages indicate that an error had occurred, with direct impact on Asterisk's functionality.
VERBOSE	Verbose-level provides additional information about asterisk functionality in run time.
DTMF	DTMF provides information about DTMF messages and information passing through the Asterisk server.



The default levels defined in AsteriskNOW for the logging are: NOTICE, WARNING, and ERROR. In order to control the levels reported, please refer to Chapter 11 for advanced logging configuration.

AsteriskNOW System Info

The system information section provides an insight to the system's health condition. While the information is fairly basic, it is an integral part of any computer system. Click on the **System Info** link of your main menu; the following screen should be observed:

```

System Information ⓘ
General  ifconfig  Resources

OS Version:
Linux asterisknow.localdomain 2.6.19.7-0.4.smp.gcc3.4.x86.i686
#1 SMP Tue May 1 12:29:52 EDT 2007 i686 i686 i386 GNU/Linux

Uptime:
16:22:40 up 2 days, 4:57, 1 user,
Load Average: 0.00, 0.00, 0.00

Asterisk Build:
Asterisk 1.4.5
Asterisk GUI-version Revision: 11425 ⓘ

Server Date & TimeZone:
Tue Sep 11 16:22:40 IDT 2007

Hostname:
asterisknow.localdomain

```

As you can see, the **System information** is separated into three sections: **General**, **ifconfig**, and **Resources**. You are currently looking at the **General** system information, providing some general information about our system. Proceed to the **ifconfig** section, showing us the configuration and statistics of our network connection.

```

System Information ⓘ
General  ifconfig  Resources

ifconfig:

eth0      Link encap:Ethernet  HWaddr 00:19:D1:29:B6:1C
          inet addr:192.168.2.122  Bcast:192.168.2.255  Mask:255.255.255
          inet6 addr: fe80::219:d1ff:fe29:b61c/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:196738 errors:0 dropped:0 overruns:0 frame:0
          TX packets:30614 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:100
          RX bytes:54314342 (51.7 Mb)  TX bytes:6168424 (5.8 Mb)
          Base address:0x30c0 Memory:30300000-30320000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:39818 errors:0 dropped:0 overruns:0 frame:0
          TX packets:39818 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:4223100 (4.0 Mb)  TX bytes:4223100 (4.0 Mb)

```

The **ifconfig** screen provides information about our connected and active network connections. Pay attention to the **errors** and **dropped** indications; if these start growing, your server may require a new network card. Proceed to the **Resources** section, which shows us the currently available resources of our server.

System Information 

General

ifconfig

Resources

Disk Usage:

Filesystem	Size	Used	Avail	Use%	Mounted on
/dev/sda2	72G	1.5G	67G	3%	/
/dev/sda1	99M	12M	83M	12%	/boot
/dev/shm	243M	0	243M	0%	/dev/shm

Memory Usage:

	total	used	free	shared	buffers	cached
Mem:	497568	354396	143172	0	37944	234704
-/+ buffers/cache:		81748	415820			
Swap:	1044216	0	1044216			

The **Resources** screen provides us with information about **Disk Usage** and **Memory Usage**. Please note that you need to keep track of these, because if your hard-drive reaches 100%, the system will stop functioning. In addition, keep track of your memory utilization, as usage of swap memory is not recommended.

AsteriskNOW Active Channels

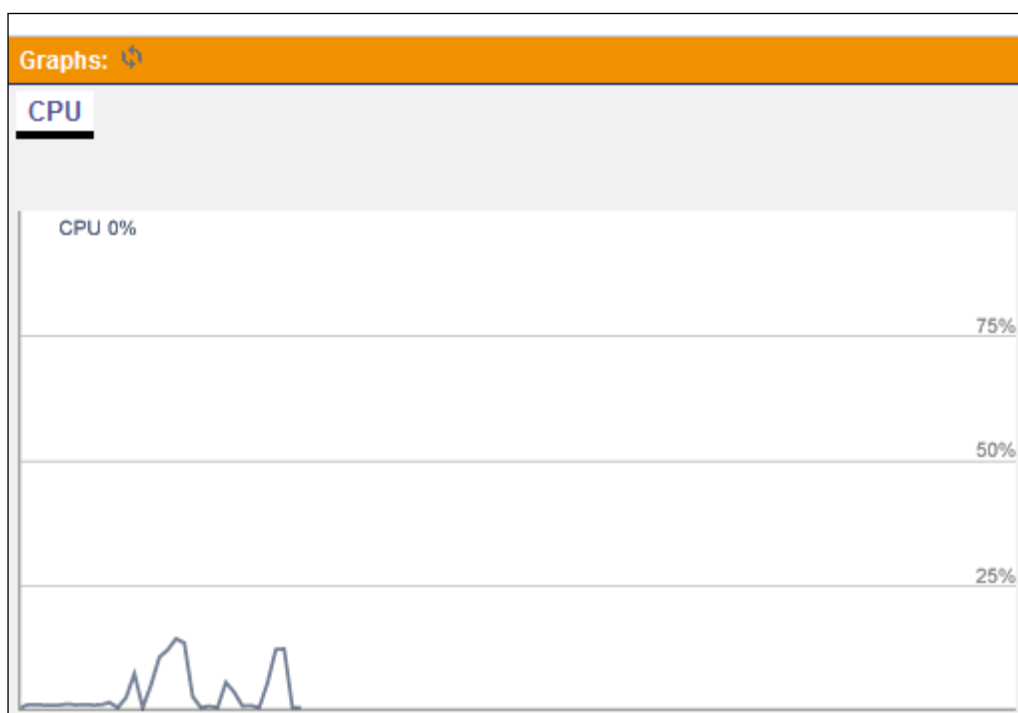
The active channels section provides an insight into the currently connected channels on the PBX (calls). Click on the **Active Channels** link of your main menu; the following will appear on your screen (this is an extract from an active system; your screen would look different):

Active Channels				
Refresh Transfer... Hangup				
Channel	State	Caller	Location	Link
SIP	unknown	Unknown	None	None
SIP/6501-08295eb0	Up	User 6501 <6501>	s@macro-trunkdial:1	IAX2/trunk_2-1
IAX2/trunk_2-1	Up	56666	None	SIP/6501-08295eb0

As you can see, there are two active channels, which are indicated as linked to one another – indicating a connected call. While in the channels view, you can transfer a currently connected channel or hang up, using the **Transfer** and **Hangup** buttons.

AsteriskNOW Graphs

The **Graphs** section provides a real-time graph of the CPU utilization of your PBX. Click on the **Graphs** link of your main menu; the following screen should be observed:



The CPU graph is generated using an AJAX automatically refreshed page, which generates an SVGZ (Scaleable Vector Graphics Z-Compression)-based image file. While this file format is natively supported by Linux Firefox, Windows-based Firefox seems to lack this support.

Summary

AsteriskNOW provides various information to its administrator, allowing the administrator to effectively monitor, manage, and debug the installation. These abilities can be extended to support additional functionality – as will be discussed in Chapter 11.

11

Hard Core AsteriskNOW

*Any sufficiently advanced technology is indistinguishable from magic.—
Arthur C. Clarke*

While AsteriskNOW is meant to serve as a simple Asterisk software appliance, bridging the gap between a novice user and an expert, it is clear that some advanced usage is available and when performed by a skilled person, will be deemed as magic by the untrained eye.

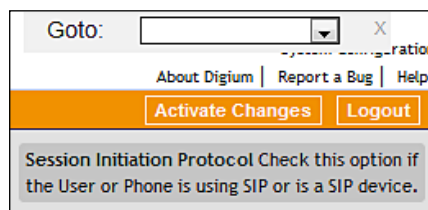
AsteriskNOW enables some advanced configuration abilities and usage that allow the administrator to go beyond the normal scope of operations. This chapter will discuss these advanced options.

AsteriskNOW Advanced Options

In the previous chapter, the various options under the **Options** section of the main menu were discussed. Go back to that menu option. Click the **Options** link of the main menu and examine the title bar. The following should be observed:



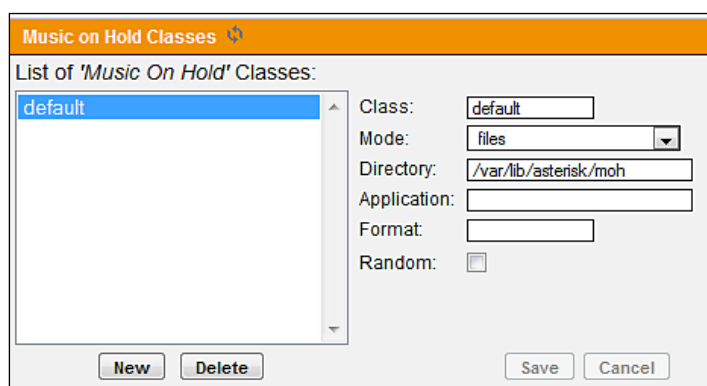
Click the **Show Advanced Options** link, and observe the change in the screen, at the top right hand of the screen:



A selection box appears, which enables some low-level control over some of AsteriskNOW's advanced configurations. The advanced configuration selection box enables the setting of the following aspects: Music on Hold, Voicemail email notification format, Global SIP, Global IAX2 settings, setting the administrative password and access to the first-time configuration wizard (the same wizard that was used during the initial configuration stage).

Music on Hold

Select the **Music On Hold** option from the select box. The following screen should be observed:



The existing default music on hold class can be modified or new classes can be created. A music on hold class consists of a class name, a mode of operation, the directory where the music on hold files are stored, and some other options. The options and their significance are as follows:

Option	Description
Class	The class name of your music on hold class.
Mode	The mode of operation for the class. The available modes are: quietmp3 : Default mode mp3 : Loud mp3nb : Unbuffered quietmp3nb : Quiet unbuffered custom : Run a custom application (useful for streaming) files : Read files from a directory in any Asterisk-supported format
Directory	The directory from which the music on hold files have to be read.
Application	Application command line for custom mode operations.

Option	Description
Format	Format of the output stream for custom mode operations; defaults to uLaw.
Random	Play the files in the directory in a random order, when mode is not set to custom .

Why Do You Need Multiple Music On Hold Classes?

In Chapter 9, one of the parameters available when defining a new queue is the queue's music on hold class. The combination of multiple music on hold classes and multiple queues can create a situation where different queues are assigned different music on hold classes. The advantage is that you can now direct your music on hold according to the queue. For example, a person waiting in the queue to talk to a sales representative of the notebook department, would surely like to listen to some interesting notebook offers, while waiting for the next available agent.

VM Email Settings

Select the **VM Email settings** option from the select box. The following screen should be observed:

Settings for Voicemail notification Emails

Template for Voicemail Emails

From asterisk

Subject

Message

Save Cancel

Template Variables: \t : TAB

{VM_NAME} : Recipient's firstname and lastname
{VM_DUR} : The duration of the voicemail message
{VM_MAILBOX} : The recipient's extension
{VM_CALLERID} : The caller id of the person who left the message
{VM_MSGNUM} : The message number in your mailbox
{VM_DATE} : The date and time the message was left

This editor enables you to create your own customized voicemail to email notification message. You can also use a few built-in Asterisk variables to add additional content to your email notification, regarding the contents of the voicemail message.

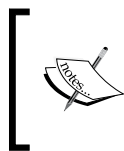
The following is an example of a customized voicemail notification message:

From	asterisk
Subject	New Voicemail from {VM_CALLERID}
Message	Dear {VM_NAME}, The PBX had received a new voicemail message for you, from caller ID {VM_CALLERID}, received at {VM_DATE} of length {VM_DUR} minutes. Kind Regards, Your Friendly AsteriskNOW PBX

Global SIP Settings

The settings appearing in Global SIP settings include a multitude of options for controlling the general SIP behavior of your AsteriskNOW PBX. There are over 80 possible options, each one capable of enabling or disabling a different aspect of the SIP protocol.

A few options are explained in the following table.



For detailed information about each of these settings, please refer to the sip.conf configuration file documentation, available at the following website: <http://www.voip-info.org/wiki/index.php?page=Asterisk+config+sip.conf>.

General SIP Settings

Option	Description
Context	A default context to send incoming SIP calls to.
Realm for digest authentication	The name of the SIP realm to be used when using digest authentication for SIP clients – defaults to Asterisk .
UDP Port to bind to	The UDP port Asterisk listens for SIP requests; defaults to 5060 .
IP Address to bind to	If your server has multiple IP numbers, specify the IP number to bind your AsteriskNOW PBX to; defaults to 0.0.0.0 (all IP addresses).
Domain	CSV list of domain your PBX renders services for; defaults to null .
Allow guest calls	Allow inbound SIP calls to be accepted from any IP address; defaults to no for security reasons.
Overlap dialling support	Enable SIP dialling support; defaults to yes .
Allow Transfers	Enable call transfers between SIP-based terminals; defaults to yes .
Enable DNS SRV lookups	Enable DNS reverse lookup for SIP terminals; defaults to yes (on most systems, a value – no – is recommended).
Pedantic	Enable pedantic checks of SIP headers; defaults to no .

Type of Service Settings

Option	Description
Max Registration/Subscription Time	The maximum time a registered SIP terminal can be registered to the PBX, before re-registering with the PBX. Defaults to 3600 seconds .
Min Registration/Subscription Time	The minimum time a registered SIP terminal will be registered to the PBX, before being removed from the PBX registry. Defaults to 60 seconds .
Time between MWI checks	Some SIP terminals enable MWI (Message Waiting Indication) notifications to be sent to them. The time indicated sets the number of seconds between each check of each SIP terminal voicemail box for new waiting messages.
Language	Default language for SIP connections; defaults to English (en) .
Enable Relaxed DTMF	Some SIP terminals may require handling of DTMF to halt normal voice operation – if so, enable this option.
RTP TimeOut	Terminate the call if no RTP (Voice/Video) had been detected on the line for this period of time; defaults to 60 seconds .
RTP HoldTimeOut	Terminate the call if no RTP (Voice/Video) had been detected on the line for this period of time and the call is currently put on hold; defaults to 300 seconds .
DTMF Mode	Default DTMF mode for SIP connections; available options are: RFC2833 , IN-BAND , INFO , or auto .

NAT Support Settings

Option	Description
Extern IP	When Asterisk or AsteriskNOW operates behind a NAT firewall, and connections from the outside world to the PBX are required, you need to configure the static NAT IP number of your AsteriskNOW server. This field holds the IP number that is assigned to your AsteriskNOW server, in accordance with your router/firewall NAT policy.
Extern Host	Same as above; however, will perform a DNS lookup on the specified hostname, and will use the resolved IP address as the external IP – useful for dynamic IP allocation with dynamic DNS resolving.
Extern Refresh	How often to refresh the Extern Host information; recommended value is 900 seconds.
Local Network Address	Local network addresses specify the internal networks directly connected to your PBX. These may include the following networks: 192.168.0.0/16 172.16.0.0/12 10.0.0.0/8 169.254.0.0/16
NAT mode	Global NAT support, affecting all service providers and extensions. The following values are available: Yes – Always ignore SIP information and assume NAT. No – Use NAT mode only according to RFC3581. Never – Never attempt NAT mode. Route – Assume NAT, don't send report. If your AsteriskNOW installation is located behind a NAT firewall/router, this should be set to Yes .
Allow RTP Reinvite	Allow for SIP RTP re-invite to occur – the default with Asterisk is Yes . When your AsteriskNOW installation is located behind a NAT firewall/router, this should be set to No .

Global IAX Settings

The settings appearing in Global IAX settings include a multitude of options, for controlling the general IAX behavior of your AsteriskNOW PBX. There are over 40 possible options, each one capable of enabling or disabling a different aspect of the IAX protocol.

A few options are explained in the following table.



For detailed information about each of these settings, please refer to the `iax.conf` configuration file documentation, available at the following website: <http://www.voip-info.org/wiki/index.php?page=Asterisk+config+iax.conf>.

General IAX Settings

Option	Description
Bind Port	The UDP port the IAX protocol will bind to; default is 4569 .
Bind Address	If your server has multiple IP numbers, specify the IP number to bind your AsteriskNOW PBX to; defaults to 0.0.0.0 (all IP addresses).
IAX1 Compatibility	Allows IAX compatibility with old IAX1 implementations. This should be enabled only if your PBX is connected to old Asterisk-based servers, prior to version 1.0.X.
No Checksums	Disable UDP checksums.
Delay Reject	If an IAX reject is required to be sent for an authentication request, delay the response. This enables increased security against IAX brute-force attacks.
ADSI	If you have ADSI-enabled phones directly connected to your PBX, you can enable support for these here.

Jitter Buffer Settings

In voice over IP (VoIP), a jitter buffer is a shared data area where voice packets can be collected, stored, and sent to the voice processor in evenly spaced intervals. Variations in packet arrival time, called jitter, can occur because of network congestion, timing drift, or route changes. The jitter buffer, which is located at the receiving end of the voice connection, intentionally delays the arriving packets so that the end user experiences a clear connection with very little sound distortion. There are two kinds of jitter buffers, static and dynamic. A static jitter buffer is hardware-based and is configured by the manufacturer. A dynamic jitter buffer is software-based and can be configured by the network administrator to adapt to changes in the network's delay. — <http://www.searchvoip.com>.

Generally speaking, a jitter buffer is a piece of software that adds small amount of delay to your voice, while at the same time allowing the voice packets assembled back at the receiving end to be assembled correctly.

Option	Description
Enable Jitter Buffer	Enable or disable the jitter buffer for IAX globally.
Force Jitter Buffer	In an ideal network, this would usually be disabled. However, as some endpoints may implement poor jitter buffering, sometimes this option needs to be enabled.
Drop Count	The drop count is the maximum number of voice packets to allow to drop (out of 100). Useful values are 3-10.
Max Jitter Buffer	The maximum available size for the jitter buffer.
Max Interpolation Frames	The maximum number of interpolation frames the jitterbuffer should return in a row.
Resync Threshold	When the jitter buffer notices a significant change in the delay between IAX frames, it will resync the jitter buffer, according to this value. Set the value to -1 to disable resynching completely.

IAX Registration Options

Option	Description
Min Reg Expire	Minimum time an IAX endpoint may request as its expiration time.
Max Reg Expire	Maximum time an IAX endpoint may request as its expiration time.
IAX ThreadCount	Establishes the number of IAX helper threads to handle I/O.
IAX Max ThradCount	The maximum number allowed of IAX helper threads.
Register	Remote IAX server registration string. This is usually performed by creating a new service provider.
Auto Kill	If you don't get ACK to your NEW within 2000ms, and Auto kill is set to yes , then you cancel the whole thing (that's enough time for one retransmission only). This is used to keep things from stalling for a long time for a host that is not available, but would be ill advised for bad connections.
Authentication Debugging	Enables or disables IAX authentication debugging appearing on the Asterisk CLI console.
Codec Priority	Defines the codec negotiation precedence; applicable values are: Caller – Consider the caller's preferred order ahead of the host's. Host – Consider the host's preferred order ahead of the caller's. Disable – Disable codec consideration. Reqonly – Same as disabled; however, only don't use consideration if the requested codec is not available. Default is Host .

Codecs Settings

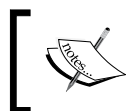
Option	Description
Disallowed Codecs	List of disallowed codecs. See Chapter 1 for more information about available codecs.
Allowed Codecs	List of allowed codecs. See Chapter 1 for more information about available codecs.

Change Password

Changes the administrative password for the admin user.

Setup Wizard

Reinitiates the initial setup wizard (as described in Chapter 2).



The following sections are for experienced Linux system administrators, or Asterisk power users wishing to gain more power over their AsteriskNOW appliance.

Gaining Root Access to Your AsteriskNOW via SSH

By default, AsteriskNOW does not enable any root access to it via SSH. However, as most Linux system administrators like to have root access to their installed servers, AsteriskNOW should not be any different. To enable root access to your AsteriskNOW server, you can either change the password via the rPath administration interface, or perform the following steps:

- **Step 1 – Enter the AsteriskNOW CLI from the console:** From the directly connected console, press the *ALT+F9* keys. This will shift your console to the AsteriskNOW CLI.

The screen should be similar to the following:

```
== Registered application 'DateTime'
app_sayunixtime.so => <Say time>
== Registered application 'SetCDRUserField'
== Registered application 'AppendCDRUserField'
== Manager registered action SetCDRUserField
app_setcdruserfield.so => <CDR user field apps>
== Registered application 'ZapScan'
app_zapscan.so => <Scan Zap channels application>
== Parsing '/etc/asterisk/codecs.conf': Found
-- codec_adpcm: using generic PLC
== Registered translator 'adpcmtolin' from format adpcm to slin, cost 1
== Registered translator 'lintoadpcm' from format slin to adpcm, cost 1
codec_adpcm.so => <Adaptive Differential PCM Coder/Decoder>
== Registered custom function CALLERID
func_callerid.so => <Caller ID related dialplan function>
== Parsing '/etc/asterisk/festival.conf': Found
== Registered application 'Festival'
app_festival.so => <Simple Festival Interface>
== Registered application 'UserEvent'
app_userevent.so => <Custom User Event Application>
== Registered custom function RAND
func_rand.so => <Random number dialplan function>
Asterisk Ready.
*CLI> █
```

- **Step 2—Modify the root user password:** From your AsteriskNOW CLI, issue the following command:

```
*CLI> !passwd root
Changing password for user root.
New UNIX password:
Retype new UNIX password:
passwd: all authentication tokens updated successfully.
*CLI>
```

At this point, the root password is set as you provisioned it.

- **Step 3—Modify SSH server configuration:** From your AsteriskNOW CLI, issue the following command:

```
*CLI> !vi /etc/ssh/sshd_config
```

The following should now appear on your screen:

```
# $OpenBSD: sshd_config,v 1.74 2006/07/19 13:07:10 dtucker Exp $
# This is the sshd server system-wide configuration file. See
# sshd_config(5) for more information.
# This sshd was compiled with PATH=/usr/local/bin:/bin:/usr/bin
# The strategy used for options in the default sshd_config shipped with
# OpenSSH is to specify options with their default value where
# possible, but leave them commented. Uncommented options change a
# default value.

#Port 22
#Protocol 2,1
Protocol 2
#AddressFamily any
#ListenAddress 0.0.0.0
ListenAddress ::

# HostKey for protocol version 1
#HostKey /etc/ssh/ssh_host_key
# HostKeys for protocol version 2
HostKey /etc/ssh/ssh_host_rsa_key
```

18.1 Top

Search for the line that says `PermitRootLogin`, verify that it is uncommented and that it is set to `yes`, as the following:

```
# Authentication:
#LoginGraceTime 2m
PermitRootLogin yes
#StrictModes yes
#MaxAuthTries 6
```

- **Step 4—Restart your SSH server:** From your AsteriskNOW CLI, issue the following command:

```
*CLI> !service sshd restart
Stopping sshd: [ OK ]
Starting sshd: [ OK ]
*CLI>
```

Now, try logging in to AsteriskNOW from an SSH client (e.g. PuTTY on windows) to access your AsteriskNOW.

The Asterisk Command-Line Interface (CLI)

If you are an experienced Linux user or network engineer, you must be familiar with the CLI concept from other appliances. Common CLI-enabled appliances include routers, switches, firewall appliances other network-attached equipment. The Asterisk CLI is no different, and provides an insight into Asterisk internal workings and current status.

To enter the Asterisk CLI, you must first SSH as root to your AsteriskNOW server, and issue the following command:

```
[root@asterisknow ~]# asterisk -vvvcgr
```



The above command indicates a certain level of verbosity using the `v` option. Additional `v` characters indicate a higher level of verbosity. It is also possible to enable debugging and other console messages, by modifying the `logger.conf` file. However, it is advisable after modifying the `logger.conf` file to change the modification to this file back, as permanent logging and debugging will cause your hard drive to fill up quickly and in some cases may degrade the performance.

The following should appear on your screen:

```
[root@asterisknow ~]# asterisk -vvvcgr
Asterisk 1.4.9, Copyright (C) 1999 - 2007 Digium, Inc. and others.
Created by Mark Spencer <markster@digium.com>
Asterisk comes with ABSOLUTELY NO WARRANTY; type 'core show warranty' for details.
This is free software, with components licensed under the GNU General Public
License version 2 and other licenses; you are welcome to redistribute it under
certain conditions. Type 'core show license' for details.
=====
== Parsing '/etc/asterisk/asterisk.conf': Found
== Parsing '/etc/asterisk/extconfig.conf': Found
Connected to Asterisk 1.4.9 currently running on asterisknow (pid = 26710)
Verbosity is at least 4
asterisknow*CLI> █
```

There are over 200 commands available via the Asterisk CLI, each one with its own set of parameters and values. Following is a list of the available CLI commands:

Command	Description
!	Executes a shell command
abort halt	Cancels a running halt
ael debug contexts	Enables AEL contexts debug (does nothing)
ael debug macros	Enables AEL macros debug (does nothing)
ael debug read	Enables AEL read debug (does nothing)
ael debug tokens	Enables AEL tokens debug (does nothing)
ael nodebug	Disables AEL debug messages
ael reload	Reloads AEL configuration
agent logoff	Sets an agent offline
agent show	Shows status of agents
agent show online	Shows all online agents
agi debug	Enables AGI debugging
agi debug off	Disables AGI debugging
agi dumphtml	Dumps a list of agi commands in HTML format
agi show	Lists AGI commands or specific help
cdr status	Displays the CDR status
console active	Sets/displays active console
console answer	Answers an incoming console call
console autoanswer	Sets/displays autoanswer
console boost	Sets/displays mic boost in dB
console dial	Dials an extension on the console
console flash	Flashes a call on the console
console hangup	Hangs up a call on the console
console mute	Disables mic input
console send text	Sends text to the remote device
console transfer	Transfers a call to a different extension
console unmute	Enables mic input
core clear profile	Clears profiling info
core set debug channel	Enables/disables debugging on a channel
core set debug	Sets level of debug chattiness

Command	Description
core set debug off	Turns off debug chattiness
core set global	Sets global dial-plan variable
core set verbose	Sets level of verbosity
core show applications	Shows registered dial-plan applications
core show application	Describes a specific dial-plan application
core show audio codecs	Displays a list of audio codecs
core show channels	Displays information on channels
core show channel	Displays information on a specific channel
core show channeltypes	Lists available channel types
core show channeltype	Gives 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	Displays config mappings (file names to config engines)
core show file formats	Displays file formats
core show file version	Lists versions of files used to build Asterisk
core show functions	Shows registered dial-plan functions
core show function	Describes a specific dial-plan function
core show globals	Shows global dial-plan variables
core show hints	Shows dial-plan hints
core show image codecs	Displays a list of image codecs
core show image formats	Displays image formats
core show license	Shows the license(s) for this copy of Asterisk
core show profile	Displays profiling info
core show switches	Shows alternative switches
core show threads	Shows running threads
core show translation	Displays translation matrix
core show uptime	Shows uptime information
core show version	Displays version info
core show video codecs	Displays a list of video codecs
core show warranty	Shows the warranty (if any) for this copy of Asterisk
database del	Removes database key/value

Command	Description
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 for a specific database key
dialplan add extension	Adds new extension into context
dialplan add ignorepat	Adds new ignore pattern
dialplan add include	Includes context in other context
dialplan reload	Reloads extensions and *only* extensions
dialplan remove extension	Removes a specified extension
dialplan remove ignorepat	Removes ignore pattern from context
dialplan remove include	Removes a specified include from context
dialplan save	Saves dialplan
dialplan show	Shows dialplan
dnsmgr reload	Reloads the DNS manager configuration
dnsmgr status	Displays the DNS manager status
dundi debug	Enables DUNDi debugging
dundi flush	Flushes DUNDi cache
dundi lookup	Looks up a number in DUNDi
dundi no debug	Disables DUNDi debugging
dundi no store history	Disables DUNDi historic records
dundi precache	Precaches a number in DUNDi
dundi query	Queries a DUNDi EID
dundi show entityid	Displays Global Entity ID
dundi show mappings	Shows DUNDi mappings
dundi show peers	Shows defined DUNDi peers
dundi show peer	Shows info on a specific DUNDi peer
dundi show precache	Shows DUNDi precache
dundi show requests	Shows DUNDi requests
dundi show trans	Shows active DUNDi transactions
dundi store history	Enables DUNDi historic records
feature show	Lists configured features

Command	Description
file convert	Converts audio file
group show channels	Displays active channels with group(s)
gtalk reload	Reload the chan_gtalk module
gtalk show channels	Shows GoogleTalk channels
h.323 reload	Reloads H.323 configuration
h323 cycle gk	Manually re-registers with the Gatekeeper
h323 hangup	Manually try to hang up a call
h323 set debug	Enables H.323 debug
h323 set debug off	Disables H.323 debug
h323 set trace	Enables H.323 Stack tracing
h323 set trace off	Disables H.323 Stack tracing
h323 show tokens	Shows all active call tokens
help	Displays help list, or specific help on a command
http show status	Displays HTTP server status
iax2 provision	Provisions an IAX device
iax2 prune realtime	Prunes a cached realtime lookup
iax2 reload	Reloads IAX configuration
iax2 set debug	Enables IAX debugging
iax2 set debug jb	Enables IAX jitterbuffer debugging
iax2 set debug jb off	Disables IAX jitterbuffer debugging
iax2 set debug off	Disables IAX debugging
iax2 set debug trunk	Enables IAX trunk debugging
iax2 set debug trunk off	Disables IAX trunk debugging
iax2 show cache	Displays IAX cached dialplan
iax2 show channels	Lists active IAX channels
iax2 show firmware	Lists available IAX firmwares
iax2 show netstats	Lists active IAX channel netstats
iax2 show peers	Lists defined IAX peers
iax2 show peer	Shows details on specific IAX peer
iax2 show provisioning	Displays IAX provisioning
iax2 show registry	Displays IAX registration status
iax2 show stats	Displays IAX statistics

Command	Description
<code>iax2 show threads</code>	Displays IAX helper thread info
<code>iax2 show users</code>	Lists defined IAX users
<code>iax2 test loss_pct</code>	Sets IAX2 incoming frame loss percentage
<code>indication add</code>	Adds the given indication to the country
<code>indication remove</code>	Removes the given indication from the country
<code>indication show</code>	Displays a list of all countries/indications
<code>jabber debug</code>	Enables Jabber debugging
<code>jabber debug off</code>	Disables Jabber debug
<code>jabber reload</code>	Reloads Jabber configuration
<code>jabber show connected</code>	Shows state of clients and components
<code>jabber test</code>	Shows roster, but is generally used for mog's debugging.
<code>keys init</code>	Initializes RSA key passcodes
<code>keys show</code>	Displays RSA key information
<code>local show channels</code>	Lists status of local channels
<code>logger mute</code>	Toggles logging output to a console
<code>logger reload</code>	Reopens the log files
<code>logger rotate</code>	Rotates and reopens the log files
<code>logger show channels</code>	Lists configured log channels
<code>manager show command</code>	Shows a manager interface command
<code>manager show commands</code>	Lists manager interface commands
<code>manager show connected</code>	Lists connected manager interface users
<code>manager show eventq</code>	Lists manager interface queued events
<code>manager show users</code>	Lists configured manager users
<code>manager show user</code>	Displays information on a specific manager user
<code>meetme</code>	Executes a command on a conference or conferee
<code>mgcp audit endpoint</code>	Audits specified MGCP endpoint
<code>mgcp reload</code>	Reloads MGCP configuration
<code>mgcp set debug</code>	Enables MGCP debugging
<code>mgcp set debug off</code>	Disables MGCP debugging
<code>mgcp show endpoints</code>	Lists defined MGCP endpoints
<code>mixmonitor</code>	Executes a MixMonitor command
<code>module load</code>	Loads a module by name

Command	Description
module reload	Reloads configuration
module show	Lists modules and info
module show like	Lists a specific module and it's information
module unload	Unloads a module by name
moh reload	Reloads Music On Hold
moh show classes	Lists MOH classes
moh show files	List sMOH file-based classes
no debug channel	(null)
odbc show	Lists ODBC DSN(s)
originate	Originates a call
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 for a specific span on your PRI interface card
pri unset debug file	Ends PRI debug output to file
queue add member	Adds a channel to a specified queue
queue remove member	Removes a channel from a specified queue
queue show	Show status of a specified queue
realtime load	Used to print out RealTime variables.
realtime pgsql status	Shows connection information for the PostgreSQL RealTime driver
realtime update	Used to update RealTime variables.
restart gracefully	Restarts Asterisk gracefully
restart now	Restarts Asterisk immediately
restart when convenient	Restarts Asterisk at empty call volume
rtcp debug ip	Enables RTCP debugging on IP
rtcp debug	Enables RTCP debugging
rtcp debug off	Disables RTCP debugging
rtcp stats	Enables RTCP stats
rtcp stats off	Disables RTCP stats

Command	Description
rtp debug ip	Enables RTP debugging on IP
rtp debug	Enables RTP debugging
rtp debug off	Disables RTP debugging
say load	sets/shows the say mode
show parkedcalls	Lists parked calls
show queues	(null)
sip history	Enables SIP history
sip history off	Disables SIP history
sip notify	Sends a notify packet to a SIP peer
sip prune realtime	Prunes cached Realtime object(s)
sip prune realtime peer	Prunes cached Realtime peer(s)
sip prune realtime user	Prunes cached Realtime user(s)
sip reload	Reloads SIP configuration
sip set debug	Enables SIP debugging
sip set debug ip	Enables SIP debugging on IP
sip set debug off	Disables SIP debugging
sip set debug peer	Enables SIP debugging on Peername
sip show channels	Lists active SIP channels
sip show channel	Shows detailed SIP channel info
sip show domains	Lists your local SIP domains
sip show history	Shows SIP dialog history
sip show inuse	Lists all in-use/limits
sip show objects	Lists all SIP object allocations
sip show peers	Lists defined SIP peers
sip show peer	Shows details on specific SIP peer
sip show registry	Lists SIP registration status
sip show settings	Shows SIP global settings
sip show subscriptions	Lists active SIP subscriptions
sip show users	Lists defined SIP users
sip show user	Shows details on specific SIP user
skinny reset	Resets Skinny device(s)
skinny set debug	Enables Skinny debugging

Command	Description
skinny set debug off	Disables Skinny debugging
skinny show devices	Lists defined Skinny devices
skinny show lines	Lists defined Skinny lines per device
sla show stations	Shows SLA Stations
sla show trunks	Shows SLA trunks
soft hangup	Requests a hangup on a given channel
stop gracefully	Gracefully shuts down Asterisk
stop now	Shuts down Asterisk immediately
stop when convenient	Shuts down Asterisk at empty call volume
stun debug	Enables STUN debugging
stun debug off	Disables STUN debugging
udptl debug	Enables UDPTL debugging
udptl debug ip	Enables UDPTL debugging on IP
udptl debug off	Disables UDPTL debugging
voicemail show users	Lists defined voicemail boxes
voicemail show zones	Lists zone message formats
zap destroy channel	Destroys a channel
zap restart	Fully restarts zaptel channels
zap show cadences	Lists cadences
zap show channels	Shows active zapata channels
zap show channel	Shows information on a channel
zap show status	Show all Zaptel cards status

A few useful commands:

Monitoring the Currently Active Channels

```
asterisknow*CLI> core show channels
Channel          Location          State  Application(Data)
SIP/6501-08219860 6501@numberplan-cust Ringing AppDial((Outgoing
Line))
SIP/6501-08215108 s@macro-stdexten:1 Ring    Dial(SIP/6501|20)
2 active channels
1 active call
```

Disconnecting an Active Call

```

asterisknow*CLI> core show channels
Channel                Location                State  Application(Data)
SIP/6501-08218ba0      6501@numberplan-cust   Ringing AppDial((Outgoing
Line))
SIP/6501-08214c38      s@macro-stdexten:1     Ring   Dial(SIP/6501|20)
2 active channels
1 active call
-- Nobody picked up in 20000 ms
-- Executing [s@macro-stdexten:2] Goto("SIP/6501-08214c38", "s-
NOANSWER|1") in new stack
-- Goto (macro-stdexten,s-NOANSWER,1)
-- Executing [s-NOANSWER@macro-stdexten:1] VoiceMail("SIP/6501-
08214c38", "6501|u") in new stack
-- <SIP/6501-08214c38> Playing 'vm-theperson' (language 'en')
-- <SIP/6501-08214c38> Playing 'digits/6' (language 'en')
-- <SIP/6501-08214c38> Playing 'digits/5' (language 'en')
-- <SIP/6501-08214c38> Playing 'digits/0' (language 'en')
-- <SIP/6501-08214c38> Playing 'digits/1' (language 'en')
-- <SIP/6501-08214c38> Playing 'vm-isunavail' (language 'en')
-- <SIP/6501-08214c38> Playing 'vm-intro' (language 'en')
asterisknow*CLI> soft hangup SIP/6501-08214c38
Requested Hangup on channel 'SIP/6501-08214c38'

```

View the Currently Configured and Connected SIP Endpoints

```

asterisknow*CLI> sip show peers
Name/username      Host                Dyn Nat ACL Port      Status
6501/6501          192.168.2.106      D           5062    Unmonitored
6500                (Unspecified)      D           0       Unmonitored
6000                (Unspecified)      D    N     0       Unmonitored
3 sip peers [Monitored: 0 online, 0 offline Unmonitored: 1 online, 2
offline

```

Explore and experiment with the usage of the Asterisk CLI, as it is one of the most comprehensive tools to learn more about Asterisk.

The Asterisk Dial-Plan Language (extensions.conf)

As you may recall, in Chapter 6 you were introduced to the `extensions.conf` file, available via the **File Editor** link, located in the main menu. The basic concepts of the `extensions.conf` scripting language are introduced in the following sections.

Configuration File Structure

The `extensions.conf` file, like all Asterisk configuration files, is built from a set of reserved contexts, representing the mandatory Asterisk configuration contexts for that file, and a set of manually defined configuration contexts, in accordance with your needs.

A context in any configuration file will be as follows:

```
[default]
exten => s,1,Answer
exten => s,2,Playback(welcome-message)
exten => s,3,Goto(context-in-include-file,s,1) ; go to context defined
in included file
:
:
```

The text appearing in the brackets, `[default]`, indicates the context name, while any line starting with the `exten` indicates an operative instruction for Asterisk. The mandatory contexts for the `extensions.conf` file are `general` and `globals`. The following is the default information contained in the `[general]` context:

[general]
static=yes
writprotect=no
clearglobalvars=no

The `[globals]` context contains global variables that can be used in any portion of the dial plan. Following is an example of the `[globals]` context, as it appears in the Asterisk `extensions.conf` file:

[globals]
CONSOLE=Console/dsp
IAXINFO=guest
TRUNK=Zap/g2
TRUNKMSD=1
voicepulse01=SIP/voicepulse01
trunk_1=Zap/g1
trunk_2=IAX2/trunk_2

At this point the basic element of the dial-plan language is introduced, the extension (designated `exten`). The dial-plan configuration is invoked according to the following logic:

- A call is intercepted by the PBX system and is directed into its appropriate context in accordance with the configuration. In general, internal call operations are handled by the default context, while inbound routing is handled by specific contexts, controlled via the `zapata.conf` file and the `extensions.conf` file.
- Asterisk will match the called number (in some terminology this may appear as DID, DDI, or DNID) and will start processing the commands associated with that called number.
- Pattern matching may be used in order to group DID numbers into a single unifying call flow.

Examine the following dial plan (this dial plan should be similar to the one you have on your AsteriskNOW PBX):

```
[default]
exten=8500,1,VoiceMailMain
exten=8500,n,Hangup
include=voicemail-custom-1
exten=7501,1,Goto(voicemail-custom-1|s|1)
exten=8000,1,MeetMe(${EXTEN}|MsI)
exten=5000,1,Queue(${EXTEN})
exten=*11,1,agentlogin()
exten=*12,1,agentcallbacklogin()
```

Examine the following section:

```
exten=8500,1,VoiceMailMain
exten=8500,n,Hangup
```

As you may notice both the lines start with the number **8500**. This means that when a call is directed to the `[default]` context and the called number is 8500, these commands will be invoked in order – this is known as the extension number. The number appearing after the extension number is called the priority. Priorities always start with the number 1, followed by consecutive numbers or the designation `n`, indicating the next step. After the priority number, the operational part of the extension line is entered, indicating the command that should be executed at that point in the dial plan.

As you may notice, the extension number may also contain special characters, such as the # and * keys. When dealing with VoIP dialing, it is also possible to replace the extension numbers with combinations of letters and/or numbers.

Extension Pattern Matching

As mentioned earlier, it is possible to perform pattern matching on dialled numbers. A pattern match sequence will always start with the underscore symbol (`_`), indicating that Asterisk is supposed to match the pattern starting from the first digit of the number. The characters Asterisk utilizes pattern matching include the following:

- `x`: Match any digit from 0-9
- `z`: Match any digit from 1-9
- `N`: Match any digit from 2-9
- `[1237-9]`: Match any digit or letter in the brackets (in this example: 1, 2, 3, 7, 8, 9)
- `.`: Match one or more characters
- `!`: Match zero or more characters immediately

For example, assume that you have a PRI connected to your PBX and your provider has allocated you a set of inbound DID numbers starting from 1212777300 up to 1212888999. Now, suppose that you would like to pattern match all the numbers in the following range – 1212777300 up to 1212778399 – so, your pattern match strings would be as follows: `_121277[78][3-9]XX` and `_1212778[012]XX`.

Special Extensions in `extensions.conf`

In addition to the regular numeric and alpha-numeric extensions, Asterisk provides a set of special extensions that are invoked automatically in accordance with the dial plan. Following is a list containing these special extension codes:

- `i` – Invalid: Invoked when a user connected to the PBX is trying to access a non-existent extension number in the context.
- `s` – Start: Invoked when no other extension can be matched, or the channel is set for immediate answer.
- `h` – Hangup: Invoked when a call is hung up for any reason.
- `t` – Timeout: Invoked upon the expiration of a defined dial-plan timeout.
- `T` – AbsoluteTimeout: invoked upon the expiration of an Absolute Timeout for the extension or context.

- `o`—Operator: Invoked by a user when pressing 0 while within the Voicemail application, asking to be redirected to the operator.
- `a`—Voicemail break: Invoked when a user presses the * during a voicemail greeting.
- `failed`: Invoked upon the failure of an automated dial-out call.
- `fax`: Invoked upon the detection of a fax on a Zap-based channel.
- `talk`: Invoked via the BackgroundDetect application.



More detailed information about the Asterisk dial-plan language can be obtained at the following website: <http://www.voip-info.org/wiki/index.php?page=Asterisk+config+extensions.conf>.

The Asterisk Configuration Directory

Asterisk stores all its configuration information in the `/etc/asterisk` directory. Following is a list of the most common configuration files, with information about their contents. The following information is an extract from the Asterisk wiki, located at <http://www.voip-info.org>.

`agents.conf`: Configure agent channels

`h323.conf`: Configure H323 channels

`iax.conf`: Configure IAX channels

`mgcp.conf`: Configure MGCP channels

`modem.conf`: Configure Modem channels (for ISDN, not for modems!)

`phone.conf`: Configure phone channels (Linux Telephony devices)

`sip.conf`: Configure SIP channels

`sip_notify.conf`: Configure SIP NOTIFY messages

`skinny.conf`: Configure Skinny channels (Cisco SCCP)

`zapata.conf`: Configure Zap channels (Digium cards)

`extensions.conf`: The Dialplan

`extensions.ael`: The Asterisk Extensions Language

`alarmreceiver.conf`: AlarmReceiver configuration

`enum.conf`: EnumLookup configuration
`dundi.conf`: DUNDiLookup configuration
`festival.conf`: Festival configuration
`indications.conf`: Playtones tone definitions
`meetme.conf`: MeetMe conference configuration
`musiconhold.conf`: MusicOnHold configuration
`queues.conf`: Queue configuration
`voicemail.conf`: VoiceMail configuration
`features.conf`: Call Parking and other features
`logger.conf`: Configuration of what to log and where to log it
`manager.conf`: Configuration of the Asterisk manager API
`modules.conf`: Configuration of Asterisk module loading
`rtp.conf`: Configuration of RTP ports for media
`say.conf`: Internationalised numbers and dates (new in Asterisk v1.4)
`users.conf`: Generate a "user" (phone, dial plan, and just about everything)



More detailed information about the Asterisk dial-plan language can be obtained at the following website: <http://www.voip-info.org/wiki/view/Asterisk+config+files>

Summary

While the web administration front end enables the novice a high level of control over Asterisk, an experienced system administrator can achieve a higher level of control over the installed AsteriskNOW appliance and Asterisk in general. While most installations will not require this form of control, it is a good practice to get acquainted with these abilities and features, as you may require them in the future.

12

Where to from Here?

To know the road ahead, ask those coming back. —Chinese Proverb

Congratulations, you have successfully installed and executed the AsteriskNOW open-source PBX appliance. You now have a fully running PBX system and are on your way of completing your office PBX installation. At this point of time, some advanced Asterisk topics are introduced. These topics mainly cover developer information, so if you are not a developer, you can easily skip this chapter.

Beyond the Dial Plan—Asterisk Gateway Interface (AGI)

Asterisk Gateway Interface, commonly referred to as AGI, is a scripting methodology for Asterisk. Based upon a similar methodology to Common Gateway Interface (CGI), AGI utilizes `stdin/stdout` streams to provide a communication path between Asterisk and an external logic. AGI scripts are commonly used to extend beyond the functionality of the dial plan, allowing a developer to combine highly complex logic structures and external operations with internal Asterisk applications and dial-plan logic.

While a tutorial for AGI developers is beyond the scope of this book a few simple AGI examples in a simple scripting language are provided. The following examples utilize the PHP scripting language, as it is ideal for fast prototyping and my experience has shown that writing AGI scripts with PHP provides highly stable code, which is easily scalable.

AGI Execution Environment

Invoking an AGI script is performed from the Asterisk dial plan, using the following command structure:

```
Exten => _85XX,n,AGI(someagi.php|arg1|arg2|...)
```

All AGI scripts should reside in the `/var/lib/asterisk/agi-bin` directory and should have proper execution permissions, in accordance to the ownership under which Asterisk is executed.



Please note that the AGI is executed with the same permissions and ownership as the Asterisk process; thus, if your Asterisk is executed as root, your AGI scripts will be executed as root—so be careful.

Upon execution, Asterisk will provide your AGI script with some preliminary information, available directly from the Asterisk channel:

- `agi_request`: The agi filename
- `agi_channel`: The originating channel (your phone)
- `agi_language`: Typically "en"
- `agi_type`: The originating channel type e.g. "sip" or "zap"
- `agi_uniqueid`: A unique ID for the call
- `agi_callerid`: The caller ID e.g. Joe Soap <1234>
- `agi_context`: Origin context
- `agi_extension`: The called number
- `agi_priority`: The priority it was executed as in the dial plan
- `agi_accountcode`: Account code of the origin channel e.g. joesoap1
- `agi_calleridname`: The caller name e.g. Joe Soap
- `agi_callingpres`: The presentation for the caller ID in a Zap channel
- `agi_callingani2`: The number which is defined in ANI2 (only for PRI Channels)
- `agi_callington`: The type of number used in PRI channels
- `agi_callingtns`: An optional 4 digit number (Transit Network Selector) used in PRI channels
- `agi_dnid`: The dialed number ID
- `agi_rdnis`: The referring DNIS number
- `agi_enhanced`: The flag value is 1.0 if started as an EAGI script

AGI Example in PHP

Following is an example containing a PHP script compatible with the AGI interface.

```
#!/usr/bin/php -q
<?php

// Prepare the PHP environment to handle output buffer correctly
ob_implicit_flush(true);
set_time_limit(6);

// Prepare the STDIN stream for processing
$in = fopen("php://stdin","r");

// Prepare the log file for errors
$stdlog = fopen("/var/log/asterisk/my_agi.log", "w");
// toggle debugging output (more verbose)
$debug = false;
// Do function definitions before we start the main loop
function read() {
    global $in, $debug, $stdlog;
    $input = str_replace("\n", "", fgets($in, 4096));
    if ($debug) fputs($stdlog, "read: $input\n");
    return $input;
}
// Write an entry to the log file
function errlog($line) {
    global $err;
    echo "VERBOSE \"$line\"\n";
}
// Write a line to stdout
function write($line) {
    global $debug, $stdlog;
    if ($debug) fputs($stdlog, "write: $line\n");
    echo $line."\n";
}

// parse agi headers into array
// usually, these headers have no real usage within an AGI
// script, however, we must read them in order to start
// processing the script.
while ($env=read()) {
    $s = split(":", $env);
    $agi[str_replace("agi_", "", $s[0])] = trim($s[1]);
    if (($env == "") || ($env == "\n")) {
        break;
    }
}
// Main AGI Script body
```

```
echo "VERBOSE \"Here we go!\" 2\n";
read();
errlog("Call from ".$agi['channel']." - Calling phone");
read();
// Say the digits 2 and then 2 again
// X is the escape digit. As X is not DTMF, no exit is possible
write("SAY DIGITS 22 X");
read();
// Say the number 22
// X is the escape digit. As X is not DTMF, no exit is possible
write("SAY NUMBER 2233 X");
read();
// clean up file handlers etc.
fclose($in);
fclose($stdlog);
exit;
?>
```

As you can see from the above example, AGI scripts interact with Asterisk via the Stdin/Stdout interface. This means the following:

- AGI commands are sent to Asterisk via the Stdout stream.
- The AGI script receives variables from Asterisk via the Stdin stream.

AGI is a simple programming API consisting of thirty-seven commands. Following is a list of the AGI programming API, as documented in the Asterisk source code and in community documentation available at <http://www.voip-info.org>.

AGI Programming API Functions

Command	Description
answer	Asserts answer
channel status	Returns status of the connected channel
control stream file	Sends the given file allowing playback to be controlled by the given digits, if any
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
get data	Gets data on a channel

Command	Description
get full variable	Gets a channel variable, but understands complex variable names and built-in variables
get option	Behaves similar to Stream file but used with a timeout option
get variable	Gets a channel variable
hangup	Hang up 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
say date	Says a date
say datetime	Says a formatted date and time
say digits	Says a given digit string
say number	Says a given number
say phonetic	Says the given character string.
say time	Says the time
send image	Sends images to channels supporting it
send text	Sends text to channels supporting it
set autohangup	Auto hangup channel after specified time
set callerid	Sets callerID for the current channel
set context	Sets channel context
set extension	Changes channel extension
set music	Enables/Disables Music on hold generator, for example "Set Music On default"
set priority	Prioritizes the channel
set variable	Sets a channel variable
stream file	Sends audio file on channel
tdd mode	Activates TDD mode on channels supporting it to enable communication with TDDs
verbose	Logs a message to the asterisk verbose log
wait for digit	Waits for a digit to be pressed

To obtain full information about the functionality of each of the above commands, including its input and output values and parameters, please refer to the Asterisk CLI `show agi` command. The following example shows how to use the `show agi` command:

```
asterisknow*CLI>
asterisknow*CLI>
asterisknow*CLI>
asterisknow*CLI>
asterisknow*CLI>
asterisknow*CLI>
asterisknow*CLI>
asterisknow*CLI>
asterisknow*CLI>
asterisknow*CLI>
asterisknow*CLI>
asterisknow*CLI> show agi say number
Usage: SAY NUMBER <number> <escape digits>
       Say a given number, returning early if any of the given DTMF digits
       are received on the channel. Returns 0 if playback completes without a digit
       being pressed, or the ASCII numerical value of the digit if one was pressed or
       -1 on error/hangup.
asterisknow*CLI>
asterisknow*CLI>
asterisknow*CLI>
asterisknow*CLI>
asterisknow*CLI>
asterisknow*CLI>
asterisknow*CLI>
asterisknow*CLI>
```

AGI Programming Libraries

As with most APIs available to the public, the AGI programming API has a multitude of pre-packaged libraries, readily available under various open-source or proprietary licenses. A few of these libraries and their respective license types are as follows:

Programming Language	Name and URL	License
PHP	PHPAGI http://phpagi.sourceforge.net/	LGPL
PERL	Asterisk-perl http://asterisk.gnuinter.net/	The "Artistic License"
RUBY	Adhearsion http://adhearsion.com/	LGPL
JAVA	Asterisk-JAVA http://asterisk-java.org/	Apache License, Version 2.0
C	CAGI http://sourceforge.net/projects/cagi/	LGPL



Advanced users could read about FastAGI, as a means of executing AGI-type script from an external source, over the network.

Asterisk Manager Interface (AMI)

The **Asterisk Manager Interface (AMI)** allows a remote network client to connect to a pre-defined TCP port on the Asterisk server, authenticate itself, and then gather statistics and/or initiate various operations on the Asterisk server. AMI allows for external programmatic logic to interact with an Asterisk server, without actually being part of the Asterisk server like in dial-plan AGI script.

AMI is based upon three types of messages: events, actions, and responses. Events and responses are automatically invoked by the AMI server, while actions are invoked from the client side, usually to be followed by a response. The general format of an AMI action is as follows:

```
Action: <action type><CRLF>
<Key 1>: <Value 1><CRLF>
<Key 2>: <Value 2><CRLF>
...
Variable: <Variable 1>=<Value 1><CRLF>
Variable: <Variable 2>=<Value 2><CRLF>
...
<CRLF>
```

Note that the `Action:` identification always precedes the rest of the action block. Available actions are as follows:

Action	Description
AbsoluteTimeout	Sets Absolute Timeout
ChangeMonitor	Changes monitoring file of a channel
Command	Executes command
Events	Controls event flow
ExtensionState	Checks extension status
GetVar	Gets a channel variable
Hangup	Hangs up channel
IAXpeers	Lists IAX peers
ListCommands	Lists available manager commands
Logoff	Logs off manager

Action	Description
MailboxCount	Checks Mailbox message count
MailboxStatus	Checks Mailbox
Monitor	Monitors a channel
Originate	Originates a call
ParkedCalls	Lists parked calls
Ping	Pings
QueueAdd	Adds agent to Queue
QueueRemove	Removes agent from Queue
Queues	Lists Queues
QueueStatus	Queue status
Redirect	Redirects a channel
SetCDRUserField	Sets the CDR UserField
SetVar	Sets channel variable
Status	Status of a specific channel
StopMonitor	Stops monitoring a channel
ZapDialOffhook	Dials over Zap channel while offhook
ZapDNDOff	Toggle Zap channel Do Not Disturb status OFF
ZapDNDon	Toggle Zap channel Do Not Disturb status ON
ZapHangup	Hangs up Zap Channel
ZapTransfer	Transfers Zap Channel
ZapShowChannels	Shows Zap Channels
AgentCallbackLogin	Sets an agent as logged in by callback
AgentLogoff	Sets an agent as no longer logged in
Agents	Lists agents and their status
DBGet	Gets DB entry
DBPut	Puts DB entry
QueuePause	Makes a queue member temporarily unavailable
SIPshowPeer	Shows SIP peer (text format)
PlayDTMF	Plays DTMF signal on a specific channel
UpdateConfig	Updates a configuration file; used mainly by AJAM/ Asterisk-GUI

Most AGI development libraries also include an AMI development library that can be used with AGI scripts.

For more information about the Asterisk Manager Interface, please refer to the following website:

<http://www.voip-info.org/wiki/view/Asterisk+manager+API>

Asynchronous JavaScript Asterisk Manager (AJAM)

One of the pitfalls of using AMI is that it requires the skills of an advanced-level programmer. Sometimes, a web developer needs to connect to AMI and initiate actions with it. AJAM provides a web-based AMI interface via an Asterisk-embedded static web server, where web developers can connect to achieve a similar functionality to working directly with AMI.

AJAM is a fairly new technology and hasn't been utilized by most programming libraries. However, AsteriskNOW makes extensive usage of AJAM; in fact, most of AsteriskNOW's web functions are based upon a mixture of web design and AJAM operations especially the **Activate Changes** button.

For more information about AJAM, please refer to the following website:

<http://www.voip-info.org/wiki/view/Aynchronous+Javascript+Asterisk+Manager+%28AJAM%29>

Ideas and Mesh-ups

At this point, you may have some ideas about the developments you'd like to make with Asterisk in general and AsteriskNOW in particular. Remember that any mesh-up is possible, as long as you keep your mesh simple. It is one thing to mesh your PBX system with your home's security system, and a completely different thing to mesh your PBX with your refrigerator.

Here are some mesh-ups for you to think about:

Voice-Enabled Network Monitoring

Build a mesh of Nagios (<http://www.nagios.org>) an open-source network and service monitoring environment and AsteriskNOW. Create a way for Nagios to alert its defined alert contacts via a phone call, not just an email or pager message.

Voice-Enabled Intrusion Detection

Build a mesh of Snort (<http://www.snort.org>) an open-source network intrusion detection system and AsteriskNOW. Create a way for Snort to alert the network administrator upon the detection of certain hacking attempts, thus enabling rapid response to security issues – Internet Services Providers would love this one!

Voice-Enabled Attendance Clock and Proximity

Build an attendance clock for your company, based upon an AsteriskNOW PBX. Replace your existing punch clock or set of funky swipe cards. Use the employee's cellular phone as a means of identifying their request to open a door, by having them dial a door code, or by using a phone next to the door to identify them.

DUNDi—Distributed Universal Number Discovery

DUNDi™ is a peer-to-peer system for locating Internet gateways to telephony services. Unlike traditional centralized services (such as the remarkably simple and concise ENUM standard), DUNDi is fully-distributed with no centralized authority whatsoever. — <http://www.dundi.com>.

DUNDi is not a VoIP Protocol, nor is it a VoIP signaling protocol like SIP, IAX2, or H.323. DUNDi is a number discovery algorithm based upon peer-to-peer technologies, enabling fast and highly available number discovery functionality. DUNDi is an open-source project released under the terms of the GPL.

DUNDi is considered a replacement for the well established ENUM technology; however, in order to understand how it can replace it, we need to understand what each of these technologies are. DUNDi and ENUM are an integral part of any Asterisk installation and AsteriskNOW is no exception. However, its configuration is not available from the GUI, nor is it a simple one. This section does not teach how to configure DUNDi on ones AsteriskNOW appliance; however, it introduces the difference and capabilities of these technologies.

ENUM

ENUM utilizes existing DNS infrastructure to provide a methodology of translating e164 numbers to VoIP calling presentation. For example, an ENUM lookup may be performed to resolve the number 12127773456 to the SIP URL `moviephone@newyorkcity.us`. The main issue with ENUM is that it is based on a traditional hierarchical methodology, and thus requires creating a complete list of all ENUM resolvable numbers in the ENUM repository. Another issue is that ENUM doesn't provide any form of security or access control, thus anyone who has access to the DNS will be able to perform ENUM lookups with it.

DUNDi

DUNDi employs a peer-to-peer operational methodology in which each node in the DUNDi network is capable of learning what e164 numbers are connected to which node, and how to route calls within the DUNDi network efficiently.

A network of trust is defined between the nodes making up the DUNDi network. When a client connected to a node wants to call up another client, connected to another node, the directly connected node will ask its neighbors where the target client can be reached. If one of the nodes has the answer, it will answer the directly connected node, and the answer is then cached for later usage. If the neighbors are unable to answer, they will query their neighbors, and so on. Once an answer is available, all the servers in the line will cache the response; thus, the entire network now has learned where the target client is.

DUNDi enables faster route processing and call routing. This is due to the fact that the entire network learns where each client is located over the course of time, thus making every routing decision in the future shorter, in terms of time and resource considerations.

Another aspect that is immediately derived from the usage of DUNDi is that there is no single point of failure. While a node may fail, when it converges back into the network, its ability to route calls isn't impaired and it will re-learn the entire routing table quickly.

For a very simplistic example of how DUNDi works, here's an extract from the DUNDi white paper:

Russell's PC: "Hey, do you know where I can find 12565551212?"
Sara's PC: "I'm not sure. Hey Andy's PC, do you know where Russell's PC can find 12565551212?"
Andy's PC: "Not off the top of my head. Hey Beth's PC, do you know where Russell's PC can find 12565551212?"
Beth's PC: "Yes, you can find it at sip:12565551212@hsv.com."
Andy's PC: "Beth's PC says you can find it at sip:12565551212@hsv.com."
Sara's PC: "Beth's PC says you can find it at sip:12565551212@hsv.com."
Russell PC: "Thanks!"
Tom's PC: "Hey, do you know where I can find 12565551212?"
Sara's PC: "Beth's PC just said you can find it at sip:12565551212@hsv.com."
Tom's PC: "Thanks!"

The DUNDi white-paper and relevant information can be obtained from the DUNDi website, located at <http://www.dundi.com>.

Summary

As you can see, Asterisk is more than a PBX—it is a telephony Swiss army knife. In the hands of a skilful administrator, Asterisk and AsteriskNOW can be shaped and moulded into any form of application or system.

A

Jargon Buster

Over the course of this book, you've been introduced to a multitude of new acronyms and technical terms. The following pages give a quick summary of these terms, with a brief description.

Term	Description
Agent	A user on a PBX that is assigned to a queue
Asterisk	The open-source PBX
B4XXP	Quad Span BRI interfaces with hardware echo cancellation
Call Parking	Park a call on a specific extension, to have it picked up by another user or extension
Circuit Switching	Circuit switching refers to the methodology of inter-connecting end terminals before actual information can traverse between the end terminals.
CLI	Command Line Interface
Codec	Voice Encoder Decoder
Context	A group of Asterisk configuration directives, grouped into a single configuration unit
CTI	Computer Telephony Integration (usually refers to the Asterisk Manager in the Asterisk world)
DID/DDI	Direct Inward Dialing
DTMF	Dual Tone Multi Frequency (simply put, the tones generated by your telephone keypad)
DUNDi	Distributed Universal Number Discovery
ENUM	A number discovery methodology, roughly based on DNS
FXO	Foreign Exchange Office – The output jack of your analog phone line
FXS	Foreign Exchange Station – The input jack on your analog phone
G711	64kbps voice codec

Term	Description
G723.1	Voice codec capable of compressing to 5.3kbps or 6.3kbps
G726	Voice codec capable of compressing to 16kbps, 24kbps or 32kbps
G729	Voice codec capable of compressing to 8kbps
GSM	Voice codec capable of compressing to 13kbps
H323	The H.323 VoIP signaling protocol
IAX2	Inter-Asterisk Exchange Protocol
iLBC	Voice codec capable of compressing to 13.3kbps or 15.2kbps
ISDN	Integrated Services Digital Network
ISDN BRI	ISDN Basic Rate Interface—2 * 64kbps + 16kbps
ISDN PRI	ISDN Primary Rate Interface—30 * 64kbps + 16kbps
IVR	Interactive Voice Response
Jitter Buffer	A software element whose responsibility is to overcome network jitters and latency that would cause disruption in media transmission
MeetMe	The Asterisk conferencing application
MGCP	Media Gateway Control Protocol
NAT	Network Address Translation
PAT	Port Address Translation
PSTN	Plain Simple Telephony Network/Plain Stupid Telephony Network
Queue	A group identifying several users of a PBX that are supposed to ring together, according to a pre-defined strategy, while keeping track of what agent is or was talking before
Ring Group	A group identifying several connected phones that are supposed to ring together, according to a pre-defined strategy
RTP	Real Time Protocol
SIP	Session Initiation Protocol
SIP Proxy	An appliance (either hardware or software) serving as a SIP registrar and server
Speex	Variable codec compression between 2.15kbps to 22.4kbps
SS#7	Signaling System Number Seven
TC400B	Codec transcoder board
TDM2400	A 24-port analog interface card, with exchangeable modules
TDM400	A four-port analog interface card, with exchangeable modules
TDM800	An 8-port analog interface card, with exchangeable modules
TE1XXP	Single Span E1/T1 PRI interface
TE2XXP	Dual Span E1/T1 PRI interfaces
TE4XXP	Quad Span E1/T1 PRI interfaces

B

Free World Dialup (FWD)

Free World Dialup is a publicly available VoIP network, which allows its members to make free VoIP calls from one member to another. The following steps show how you can connect to Free World Dialup using your AsteriskNOW PBX system.

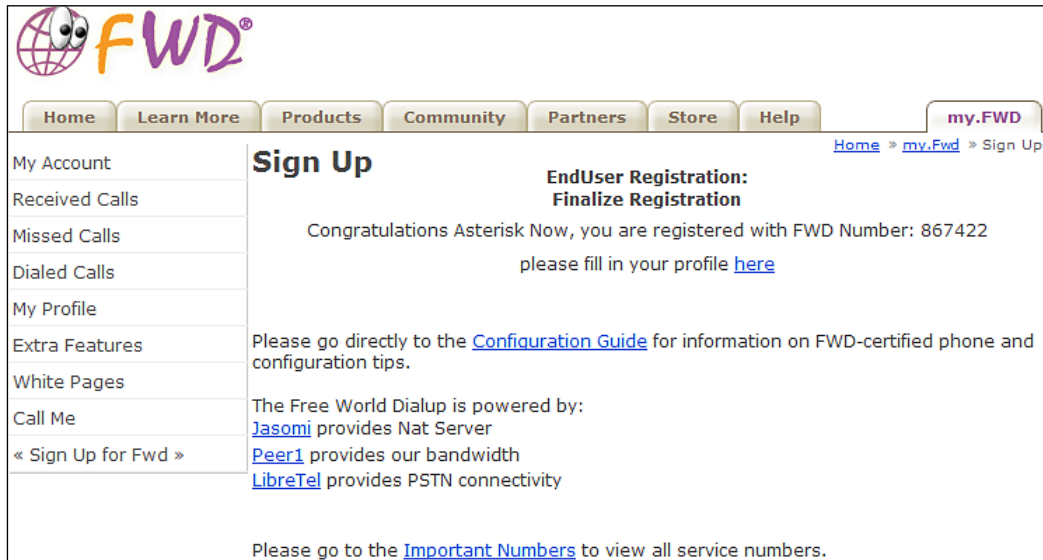
Step 1: Creating an FWD account

Log on to the FWD website, located at <http://www.fwdnet.net>, and click the **Get Started** link. You will be redirected to a sign up, and upon completion you will receive your FWD user identification and password.



The screenshot shows the FWD website homepage. At the top, there is a navigation bar with links: Home, Learn More, Products, Community, Partners, Store, Help, and my.FWD. A status bar in the top right corner indicates "32,630 FWD users online" and "FWD is running..>". The main content area features a large orange banner on the left with the text "FWD lets you talk . . . for Free . . . for Good." and a globe icon. To the right of the banner is a sidebar with links: Get Started, Partners, Peering, Network2, FWD Certified, and Forums. The main content area also includes a "Welcome to FWD" section with text about the network's history and future plans. Below the banner, there is a section titled "FWD enables anyone with a broadband connection to communicate over the Internet, whether it be instant messaging, voice, or video. Are you new to FWD? Want to [get started](#)?" followed by a "Check out FWD on Facebook!" section with a Facebook logo and text. A "Spotlight" section highlights "Peering" and "Looking for your call history and account settings?". At the bottom, there is a "Ditech Networks" logo and text about hardware provided by Ditech Networks. The footer contains a navigation bar with links: Home, Learn More, Products, Community, Partners, Store, Help, Contact, and Important Legal Stuff. It also includes a disclaimer: "FWD is not a telephony replacement service and cannot be used for emergency dialing" and a copyright notice: "2005 © Free World Dialup, LLC. All Rights Reserved."

Upon completing your registration, your screen should look as follows:

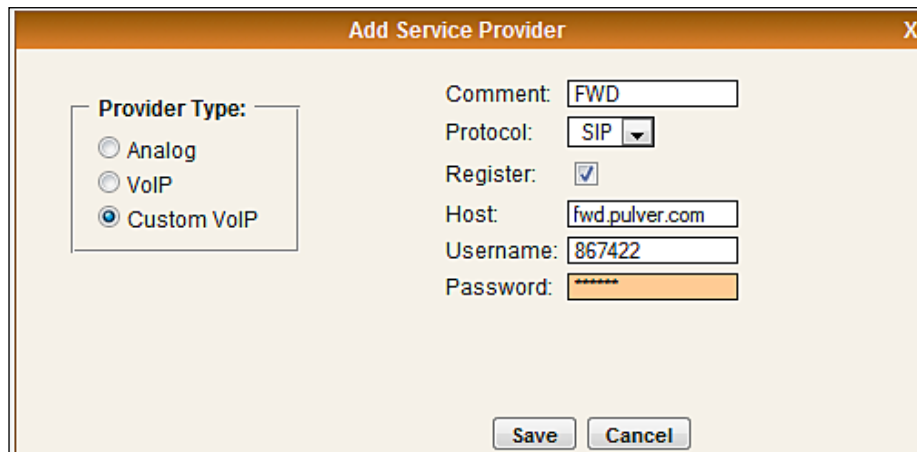


The screenshot shows the FWD website's registration completion page. At the top is the FWD logo. Below it is a navigation bar with links: Home, Learn More, Products, Community, Partners, Store, Help, and my.FWD. A breadcrumb trail reads: Home » my.Fwd » Sign Up. On the left is a sidebar menu with links: My Account, Received Calls, Missed Calls, Dialed Calls, My Profile, Extra Features, White Pages, Call Me, and « Sign Up for Fwd ». The main content area is titled 'Sign Up' and 'EndUser Registration: Finalize Registration'. It displays a congratulatory message: 'Congratulations Asterisk Now, you are registered with FWD Number: 867422' and asks the user to 'please fill in your profile here'. Below this, it provides a link to the 'Configuration Guide' for information on FWD-certified phones. It also lists the providers: 'The Free World Dialup is powered by: Jasomj provides Nat Server, Peer1 provides our bandwidth, and LibreTel provides PSTN connectivity'. At the bottom, it links to 'Important Numbers' to view all service numbers.

According to the above screenshot, the FWD number is **867422**. The number that you will receive will most probably be different, so write it down somewhere for later usage.

Step 2: Creating a customer service provider

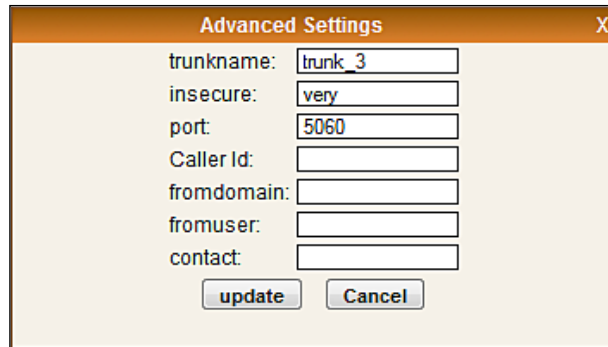
Now, go to your AsteriskNOW web management interface and add a new custom service provider. Your configuration should look similar to the following screenshot:



The screenshot shows a dialog box titled 'Add Service Provider'. It contains the following fields and options:

- Provider Type:** A group box containing three radio buttons: ☐ Analog, ☐ VoIP, and ☒ Custom VoIP.
- Comment:** A text field containing 'FWD'.
- Protocol:** A dropdown menu showing 'SIP'.
- Register:** A checkbox that is checked.
- Host:** A text field containing 'fwd.pulver.com'.
- Username:** A text field containing '867422'.
- Password:** A text field containing '*****'.
- Buttons:** 'Save' and 'Cancel' buttons at the bottom.

At this point, your service provider list should contain the "Custom -FWD" custom VoIP service provider. You now need to edit the advanced properties of this service provider, so click the **Options** combo box and then select **Advanced**. The following screen should be observed:



The "Advanced Settings" dialog box contains the following fields and buttons:

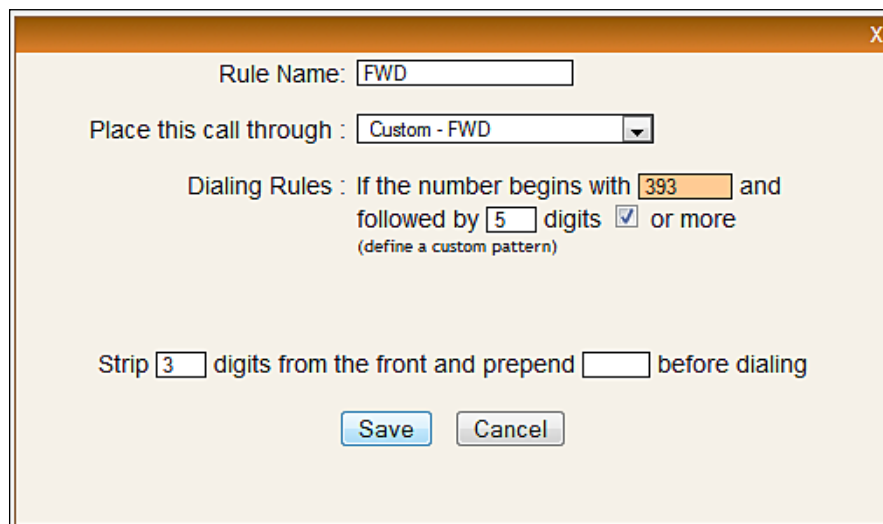
Field	Value
trunkname:	trunk_3
insecure:	very
port:	5060
Caller Id:	
fromdomain:	
fromuser:	
contact:	

Buttons: update, Cancel

Add the keyword, **very**, to the item indicated by the **insecure** option, indicating that you allow the FWD network to send calls to your AsteriskNOW without performing authentication—a fairly normal practice.

Step 3: Setting up an outbound route

Now you should create a Calling Rule, according to the following example:



The "Calling Rule" dialog box contains the following fields and buttons:

Field	Value
Rule Name:	FWD
Place this call through :	Custom - FWD
Dialing Rules :	If the number begins with 393 and followed by 5 digits or more (define a custom pattern)
Strip	3 digits from the front and prepend before dialing

Buttons: Save, Cancel

Step 4: Activating changes and testing

Now, activate your changes by clicking the **Activate Changes** button, then, pick up one of your connected phones and dial the following number: 393847821. If you are lucky, I will be at home and I'll pick up the phone and answer you, if not, you can always leave a message on my home AsteriskNOW PBX system.

C AsteriskNOW for Service Providers

AsteriskNOW provides a facility for a service provider to add its own custom-built service provider entry, which would be available via the service provider creation wizard.

The facility is available via the `/etc/asterisk/providers.conf` file. The file simply describes the available service providers and their associated information, required to make them work. The following example shows how to define Free World Dialup as a new built-in service provider in your AsteriskNOW PBX:

```
[fwd]
providername = FWD
providerlogo = http://www.freeworlddialup.com/img/fwdcomm.jpg
protocol = sip
providerdesc = Free World Dialup is a free VoIP network, allowing its
members to carry free calls from one to another.
hasiax = no
hassip = yes
registeriax = no
registersip = yes
host = fwd.pulver.com
trunk_username = freeworlddialup
```

Once you have saved the file with the bold line added in to it, reload you web GUI and try creating a new service. You will notice that you now have a new built-in service provider called **FWD**, as seen in the following screenshot:

Add Service Provider X

Provider Type:

- ☐ Analog
- ☒ VoIP
- ☐ Custom VoIP

Provider:

- Voicepulse (IAX)
- IAXTEL
- FWD**
- Voicepulse (SIP)



Free World Dialup is a free VoIP network, allowing its members to carry free calls from one to another.

Username:

Password:

Index

A

AGI

- about 161
- AGI programming API Functions 164, 166
- AGI script 162
- example in PHP 163, 164
- execution environment 162
- programming libraries 166

AJAM 169

AMI

- about 167
- AMI action 167, 169

analog interface cards

- TDM400P 27
- TDM800P 27
- TDM2400P 27

Asterisk

- dual license 20
- overview 19
- passthrough 19
- transcoding 19

Asterisk CLI

- about 146
- commands 147, 155

Asterisk codec

- availability 18
- compatibility 18

Asterisk Command Line Interface.

See Asterisk CLI

Asterisk dial-plan language

- configuration file structure 156-158
- configuration file structure, pattern matching 158
- exten 157
- extensions.conf 155

- extensions.conf, special extension codes 158, 159

- extensions.conf, special extensions 158

Asterisk Gateway Interface. *See AGI*

Asterisk Manager Interface. *See AMI*

AsteriskNOW

- active channels 132
- advanced options 135
- Asterisk CLI 146
- Asterisk configuration directory 159
- Asterisk dial-plan language 155
- Asterisk Logs 129
- backup 128, 129
- dial-plan file, editing 80-82
- downloading 26
- for service providers 179, 180
- general options 125
- graphs 133
- hardware requirements 26
- images 26
- installation process 29
- IVR generator 94
- root access, gaining via SSH 143, 145
- system information 130
- system information, general section 131
- system information, ifconfig section 132
- system information, resources section 132
- VoIP service providers 69

AsteriskNOW, advanced options

- change password 143
- global IAX settings 141
- global SIP settings 138
- multiple music on hold classes, need for 137
- music on hold class 136

- music on hold class, options and significance 136
 - setup wizard 143
 - VM email settings 137, 138
 - AsteriskNOW, general options**
 - agent login settings 127
 - extension options 127
 - local extension settings 126
 - AsteriskNOW, global IAX settings**
 - codecs settings 143
 - general IAX settings 141
 - IAX registration options 142
 - jitter buffer settings 142
 - AsteriskNOW, global SIP settings**
 - codecs settings 140
 - general SIP settings 139
 - NAT support settings 140
 - type of service settings 139
 - AsteriskNOW, hardware requirements**
 - add-on cards 28
 - analogue interface cards 27
 - analogue interface cards, with characteristics 27
 - digital interface cards 28
 - digital interface cards, with characteristics 28
 - TC400B card 28
 - AsteriskNOW, installation process**
 - AsteriskNOW distribution, installing 29-37
 - hardware, installing 29
 - initial configuration 37-46
 - AsteriskNOW codec**
 - availability 18
 - compatibility 18
 - AsteriskNOW extension management GUI**
 - about 50
 - analog extension configuration 54
 - new user extension, creating 51
 - new user extension configuration 53
 - user extension configuration flags 52
 - user extension configuration options 52
 - AsteriskNOW images**
 - AsteriskNOW (32-bit) 26
 - AsteriskNOW (64-bit) 26
 - AsteriskNOW (x86 LiveCD) 26
 - AsteriskNOW (x86 VMware image) 26
 - AsteriskNOW (x86 xen image) 26
 - Asynchronous JavaScript Asterisk Manager.**
See AJAM
- ## C
- call parking** 116
 - call queues**
 - defining 120
 - circuit-switched network**
 - about 9
 - elements 9
 - network diagram 9, 10
 - circuit-switched network, connectivity methods**
 - ISDN 11
 - SS7(Signalling System # 7) 10
 - circuit switching** 8
 - codecs**
 - about 17
 - G.729A 18
 - G.711 18
 - G.723.1 18
 - G.726 18
 - GSM 18
 - iLBC 18
 - Speex 18
 - conference room, defining**
 - conference password settings 115
 - conference room options 115
 - configuring 116
 - general conference options 114
 - CounterPath X-Lite**
 - about 60
 - configuring, to work with AsteriskNOW 63
 - downloading 61, 62
- ## D
- DID providers**
 - about 66
 - carrier/wholesale sector 66
 - LibreTel 66
 - retail sector 66
 - sectors 66
 - VoxBone 66
 - Digium** 26
 - Direct Inward Dialing providers.** *See* DID providers

Distributed Universal Number Discovery.

See **DUNDi**

DTMF 53

DUNDi

about 170, 171

ENUM 171

working 172

DUNDi network 171

E

ENUM 171

F

Free World Dialup. *See* **FWD**

FWD

about 175

connecting to, AsteriskNOW PBX system
175-178

customer service provider, creating 176

FWD account, creating 175

outbound rate, setting up 177

G

grey routes 67

H

H.323 14

I

IAX

NAT(Network Address Translation) 17

PAT(Port Address Translation) 17

IAX2

about 17

versus SIP and H.323 17

inbound call routing

about 85

example, range of DID numbers 89

example, single DID number 88, 89

inbound routing methodology 86

incoming call rule, adding 87

in extensions.conf 90

table, accessing 87

via DID numbers 85

via physical ports 86

with AsteriskNOW 86

inbound VoIP service providers

LibreTel, example 73

VoxBone, example 72

Integrated Services Digital Network. *See*
ISDN

Interactive Voice Response. *See* **IVR**

Inter Asterisk eXchange Protocol. *See* **IAX**

IP Call Termination providers

about 66

carrier/wholesale sector 67

grey routes 67

iBasis 67

NuFone 67

refilers 67

retail sector 67

sectors 66

Vonage 67

IP phone 49

IP Termination service provider

configuring 69

Custom VoIP Termination provider,
configuring 75

Custom VoIP Termination provider,
creating 75

routing table, configuring 75

ISDN

about 11

Foreign Exchange Office Interface 12

Foreign Exchange Station Interface 12

ISDN Basic Rate Interface 11

ISDN Primary Rate Interface 11

IVR

about 93

rules 93, 94

jitter buffer 142

L

LinkSys 941

about 55

extension, configuring to work with
AsteriskNOW PBX 57, 58

LinkSys 941 configuration GUI 56

registering with AsteriskNOW PBX 60

- user extension 6500, configuration parameters 58
- user extension 6500, configuring 58

M

MeetMe conferencing

- about 112
- administrator key presses 113
- conference call, activating 112
- new conference room , defining 114

mesh-ups

- voice-enabled network monitoring 169
- voice-enabled intrusion detection 170
- voice-enabled attendance clock and proximity 170

P

PBX

- feature 77

PBX system

- building 23
- call parking 116
- call parking features, defining 116
- connectivity 24
- connectivity, block diagram 24, 25
- devices 25
- inbound call routing 85
- inbound DID routing versus analog physical routing 85
- installation procedure outline 25
- internal connectivity 25
- IVR 93
- MeetMe conferencing 112
- PBX information 25
- queues 119
- routing rules 77
- service providers 65
- Trunk information 25
- voicemail system 110
- voice menus 94

PSTN carriers

- about 68
- connecting to PBX 68
- DID provider 68
- inbound service 68
- outbound service 68

- Termination provider 68

Q

queues

- about 119
- call queues, utilizing 121
- configuring steps 122
- general options 120
- queue options 121

queues, configuring steps

- agents, assigning to proper queue 122
- extensions, defining for each agent 122
- queues, defining 122

R

refilers

- about 67

ring group 102

routing rules

- about 78
- managing with AsteriskNOW 78

rPath appliance GUI

- about 47
- configuring 47, 48

S

service provider

- about 65
- PSTN carriers 68
- VoIP carriers 65

Session Initiation Protocol. *See* SIP

SIP

- about 14
- call scenario 16
- methods 15
- working 15

SIP Proxy 14

T

telephony

- about 7
- basics 8

telephony, basics

- circuit-switched network 9

circuit switching 8

Termination service providers

NuFone, example 74

Vonage, example 73

V

voicemail system

about 110

general options 110

message options 111

playback options 112

voice menus

about 94

configuring 105, 106

PBX extension, assigning 103, 104

steps 95, 96

voice menus, steps

DISA 96

recordings 97, 98

ring groups 101

ring groups, creating 101

ring groups, fields 102

time based rule, creating 99, 101

time based rules 99

VoIP carriers

about 65

DID providers, types 65

IP Call Termination providers 66

Termination providers, types 65

types 65

VoIP service providers, AsteriskNOW

configuring 69

Custom VoIP service providers 70

Custom VoIP service providers, options 71

VoIP signalling protocols

H.323 14

IAX2 17

MGCP 13

SIP 14



**Thank you for buying
AsteriskNOW**

Packt Open Source Project Royalties

When we sell a book written on an Open Source project, we pay a royalty directly to that project. Therefore by purchasing AsteriskNOW, Packt will have given some of the money received to the Trac project and AsteriskNOW project.

In the long term, we see ourselves and you – customers and readers of our books – as part of the Open Source ecosystem, providing sustainable revenue for the projects we publish on. Our aim at Packt is to establish publishing royalties as an essential part of the service and support a business model that sustains Open Source.

If you're working with an Open Source project that you would like us to publish on, and subsequently pay royalties to, please get in touch with us.

Writing for Packt

We welcome all inquiries from people who are interested in authoring. Book proposals should be sent to authors@packtpub.com. If your book idea is still at an early stage and you would like to discuss it first before writing a formal book proposal, contact us; one of our commissioning editors will get in touch with you.

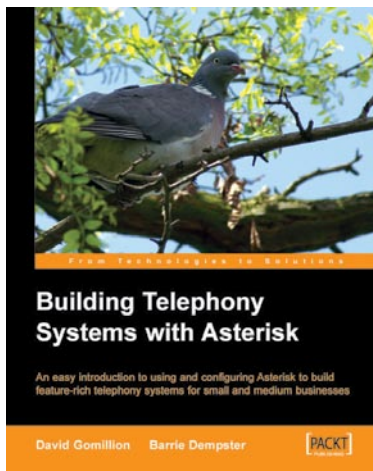
We're not just looking for published authors; if you have strong technical skills but no writing experience, our experienced editors can help you develop a writing career, or simply get some additional reward for your expertise.

About Packt Publishing

Packt, pronounced 'packed', published its first book "Mastering phpMyAdmin for Effective MySQL Management" in April 2004 and subsequently continued to specialize in publishing highly focused books on specific technologies and solutions.

Our books and publications share the experiences of your fellow IT professionals in adapting and customizing today's systems, applications, and frameworks. Our solution-based books give you the knowledge and power to customize the software and technologies you're using to get the job done. Packt books are more specific and less general than the IT books you have seen in the past. Our unique business model allows us to bring you more focused information, giving you more of what you need to know, and less of what you don't.

Packt is a modern, yet unique publishing company, which focuses on producing quality, cutting-edge books for communities of developers, administrators, and newbies alike. For more information, please visit our website: www.PacktPub.com.



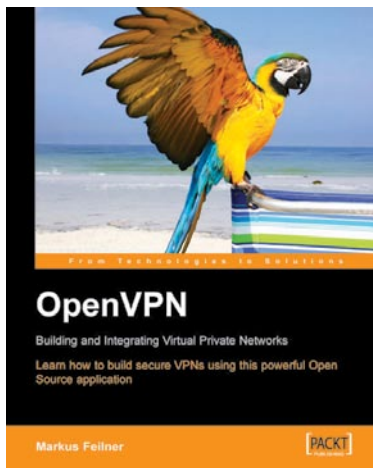
Building Telephony Systems With Asterisk

ISBN: 1-904811-15-9

Paperback: 180 pages

An easy introduction to using and configuring Asterisk to build feature-rich telephony systems for small and medium businesses.

1. Install, configure, deploy, secure, and maintain Asterisk
2. Build a fully-featured telephony system and create a dial plan that suits your needs
3. Learn from example configurations for different requirements



OpenVPN

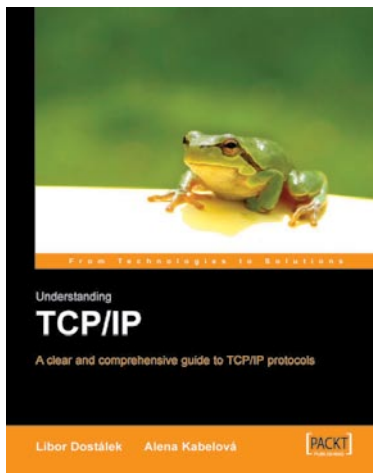
ISBN: 1-904811-85-X

Paperback: 258 pages

Learn how to build secure VPNs using this powerful Open Source application

1. Learn how to install, configure, and create tunnels with OpenVPN on Linux, Windows, and MacOSX
2. Use OpenVPN with DHCP, routers, firewall, and HTTP proxy servers
3. Advanced management of security certificates

Please check www.PacktPub.com for information on our titles

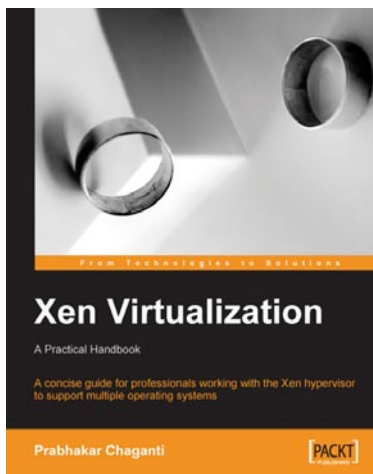


Understanding TCP/IP

ISBN: 1-904811-71-X Paperback: 462 pages

A clear and comprehensive guide to TCP/IP protocols

1. Essential reference to what's really going on at the network level
2. Covers Telnet, FTP, HTTP, SMTP, ESMTP, IMAP4, POP3, NNTP, and LDAP
3. IP4 and IP6



Xen Virtualization

ISBN: 978-1-847192-48-6 Paperback: 150 pages

A fast and practical guide to supporting multiple operating systems with the Xen hypervisor

1. Installing and configuring Xen
2. Managing and administering Xen servers and virtual machines
3. Setting up networking, storage, and encryption
4. Backup and migration

Please check www.PacktPub.com for information on our titles