

CHAPTER 1

A Telephony Revolution

*It does not require a majority to prevail,
but rather an irate, tireless minority
keen to set brush fires in people's minds.*

—Samuel Adams

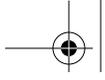
An incredible revolution is under way. It has been a long time in coming, but now that it has started, there will be no stopping it. It is taking place in an area of technology that has lapsed embarrassingly far behind every other industry that calls itself high-tech. The industry is telecommunications, and the revolution is being fueled by an open source Private Branch eXchange (PBX) called *Asterisk*TM.

Telecommunications is arguably the last major electronics industry that has (until now) remained untouched by the open source revolution. Major telecommunications manufacturers still build ridiculously expensive, incompatible systems, running complicated, ancient code on impressively engineered yet obsolete hardware.

As an example, Nortel's Business Communications Manager kludges together a Windows NT 4.0 server, a 15-year-old VXWorks-based Key Telephone Switch, and a 700-MHz PC. All this can be yours for between 5 and 15 thousand dollars, not including telephones. If you want it to actually do anything interesting, you'll have to pay extra licensing fees for closed, limited-functionality, shrink-wrapped applications. Customization? Forget it—it's not in the plan. Future technology and standards compliance? Give them a year or two—they're working on it.

All of the major telecommunications manufacturers offer similar-minded products. They don't want you to have flexibility or choice; they want you to be locked in to their product cycles.

Asterisk changes all that. With Asterisk, no one is telling you how your phone system works, or what technology you are limited to. If you want it, you can have it. Asterisk lovingly embraces the concept of standards compliance, while also enjoying the freedom to develop its own innovations. What you choose to implement is up to you—Asterisk imposes no limits.



Naturally, this incredible flexibility comes with a price: Asterisk is not a simple system to configure. This is not because it's illogical, confusing, or cryptic; to the contrary, it is very sensible and practical. People's eyes light up when they first see an Asterisk dialplan and begin to contemplate the possibilities. But when there are literally thousands of ways to achieve a result, the process naturally requires extra effort. Perhaps it can be compared to building a house: the components are relatively easy to understand, but a person contemplating such a task must either a) enlist competent help or b) develop the required skills through instruction, practice, and a good book on the subject.

VoIP: Bridging the Gap Between Traditional Telephony and Network Telephony

While Voice over IP (VoIP) is often thought of as little more than a method of obtaining free long-distance calling, the real value (and—let's be honest—challenge as well) of VoIP is that it allows voice to become nothing more than another application in the data network.

It sometimes seems that we've forgotten that the purpose of the telephone is to allow people to communicate. It is a simple goal, really, and it should be possible for us to make it happen in far more flexible and creative ways than are currently available to us. Since the industry has demonstrated an unwillingness to pursue this goal, a large community of passionate people have taken on the task.

The challenge comes from the fact that an industry that has changed very little in the last century shows little interest in starting now.

The Zapata Telephony Project

The Zapata Telephony Project was conceived of by Jim Dixon, a telecommunications consulting engineer who was inspired by the incredible advances in CPU speeds that the computer industry has now come to take for granted. Dixon's belief was that far more economical telephony systems could be created if a card existed that had nothing more on it than the basic electronic components required to interface with a telephone circuit. Rather than having expensive components on the card, Digital Signal Processing (DSP)* would be handled in the CPU by software. While this would impose a tremendous load on the CPU, Dixon was certain that the low cost of CPUs relative to their performance made them far more attractive than

* The term DSP also means Digital Signal Processor, which is a device (usually a chip) that is capable of interpreting and modifying signals of various sorts. In a voice network, DSPs are primarily responsible for encoding, decoding, and transcoding audio information. This can require a lot of computational effort.



expensive DSPs, and, more importantly, that this price/performance ratio would continue to improve as CPUs continued to increase in power.

Like so many visionaries, Dixon believed that many others would see this opportunity, and that he merely had to wait for someone else to create what to him was an obvious improvement. After a few years, he noticed that not only had no one created these cards, but it seemed unlikely that anyone was ever going to. At that point it was clear that if he wanted a revolution, he was going to have to start it himself. And so the Zapata Telephony Project was born.

Since this concept was so revolutionary, and was certain to make a lot of waves in the industry, I decided on the Mexican revolutionary motif, and named the technology and organization after the famous Mexican revolutionary Emiliano Zapata. I decided to call the card the ‘tormenta’ which, in Spanish, means ‘storm,’ but contextually is usually used to imply a big storm, like a hurricane or such.*

Perhaps we should be calling ourselves Asteristas. Regardless, we owe Jim Dixon a debt of thanks, partly for thinking this up and partly for seeing it through, but mostly for giving the results of his efforts to the open source community. As a result of Jim’s contribution, Asterisk’s Public Switched Telephone Network (PSTN) engine came to be.

Massive Change Requires Flexible Technology

The most successful key telephone system in the world has a design limitation that has survived 15 years of users begging for what appears to be a simple change: when you determine the number of times your phone will ring before it forwards to voice-mail, you can choose from 2, 3, 4, 6, or 10 ring cycles. Have you any idea how many times people ask for five rings? Yet the manufacturers absolutely cannot get their heads around the idea that this is a problem. That’s the way it works, they say, and users need to get over it.

That’s just one example—the industry is rife with them.

Another example from the same system is that the name you program on your set can only be seven characters in length. Back in the late 1980s, when this particular system was built, RAM was pretty dear, and storing those seven characters for dozens of sets represented a huge hardware expense. So what’s the excuse today? None. Are there any plans to change it? Hardly—the issue is not even officially acknowledged as a problem.

Now, it’s all very well and good to pick on one system, but the reality is that every PBX in existence suffers shortcomings. No matter how fully featured it is, something will always be left out, because even the most feature-rich PBX will always fail to anticipate

* Jim Dixon, “The History of Zapata Telephony and How It Relates to the Asterisk PBX” (<http://www.asteriskdocs.org/modules/tinycontent/index.php?id=10>).

the creativity of the customer. A small group of users will desire an odd little feature that the design team either did not think of or could not justify the cost of building, and, since the system is closed, the users will not be able to build it themselves.

If the Internet had been thusly hampered by regulation and commercial interests, it is doubtful that it would have developed the wide acceptance it currently enjoys. The openness of the Internet meant that anyone could afford to get involved. So, everyone did. The tens of thousands of minds that collaborated on the creation of the Internet delivered something that no corporation ever could have.

As with many other open source projects, such as Linux and the Internet, the explosion of Asterisk was fueled by the dreams of folks who knew that there had to be something more than what the industry was producing. The strength of the community is that it is composed not of employees assigned to specific tasks, but rather of folks from all sorts of industries, with all sorts of experiences, and all sorts of ideas about what flexibility means, and what openness means. These people knew that if one could take the best parts of various PBXs and separate them into interconnecting components—akin to a boxful of LEGO bricks—one could begin to conceive of things that would not survive a traditional corporate risk-analysis process. While no one can seriously claim to have a complete picture of what this thing should look like, there is no shortage of opinions and ideas.

Many people new to Asterisk see it as unfinished. Perhaps these people can be likened to visitors to an art studio, looking to obtain a signed, numbered print. They often leave disappointed, because they discover that Asterisk is the blank canvas, the tubes of paint, the unused brushes waiting.

Even at this early stage in its success, Asterisk is nurtured by a greater number of artists than any other PBX. Most manufacturers dedicate no more than a few developers to any one product; Asterisk has scores. Most proprietary PBXs have a worldwide support team comprised of a few dozen real experts; Asterisk has hundreds.

The depth and breadth of expertise that surrounds this product is unmatched in the telecom industry. Asterisk enjoys the loving attention of old Telco guys who remember when rotary dial mattered, enterprise telecom people who recall when voicemail was the hottest new technology, and data communications geeks and coders who helped build the Internet. These people all share a common belief: that the telecommunications industry needs a *proper* revolution.*

Asterisk is the catalyst.

* The telecom industry has been predicting a revolution since before the crash; time will tell how well they respond to the *open source* revolution.

Asterisk: The Hacker's PBX

Telecommunications companies who choose to ignore Asterisk do so at their peril. The flexibility it delivers creates possibilities that the best proprietary systems can scarcely dream of. This is because Asterisk is the ultimate hacker's PBX.

If someone asks you not to use the term hacker, refuse. That term does not belong to the mass media. They stole it and corrupted it to mean "malicious cracker." It's time we took it back. Hackers built the networking engine that is the Internet. Hackers built the Apple Macintosh and the Unix operating system. Hackers are also building your next telecom system. Do not fear; these are the good guys, and they'll be able to build a system that's far more secure than anything that exists today, because rather than being constricted by the dubious and easily cracked security of closed systems, they will be able to quickly respond to changing trends in security and fine-tune the telephone system in response to both corporate policy and industry best practices.

Like other open source systems, Asterisk will be able to evolve into a far more secure platform than any proprietary system, not in spite of its hacker roots, but rather because of them.

Asterisk: The Professional's PBX

Never in the history of telecommunications has a system so suited to the needs of business been available, at any price. Asterisk is an enabling technology, and, as with Linux, it will become increasingly rare to find an enterprise that is not running some version of Asterisk, in some capacity, somewhere in the network, solving a problem as only Asterisk can.

This acceptance is likely to happen much faster than it did with Linux, though, for several reasons:

1. Linux has already blazed the trail that led to open source acceptance, so Asterisk can follow that lead.
2. The telecom industry is crippled, with no leadership being provided by the giant industry players. Asterisk has a compelling, realistic, and exciting vision.
3. End users are fed up with incompatible, limited functionality, and horrible support. Asterisk solves the first two problems; the community has shown a passion for the latter.

The Asterisk Community

One of the compelling strengths of Asterisk is the passionate community that developed and supports it. This community, led by Mark Spencer of Digium, is keenly aware of the cultural significance of Asterisk, and they are giddy about the future.

One of the more powerful side effects caused by the energy of the Asterisk community is the cooperation it has spawned among the telecommunications professionals, networking professionals, and information technology professionals who share a love for this phenomenon. While these professions have traditionally been at odds with each other, in the Asterisk community they delight in each other's skills. The significance of this cooperation cannot be underestimated.

Still, if the dream of Asterisk is to be realized, the community must grow—yet one of the key challenges the community currently faces is a rapid influx of new users. The members of the existing community, having birthed this thing called Asterisk, are generally welcoming of new users, but they've grown impatient with being asked the kinds of questions whose answers can often be obtained independently, if one is willing to put forth the time needed to research and experiment.

Obviously, new users do not fit any particular kind of mold. While some will happily spend hours experimenting and reading various blogs describing the trials and tribulations of others, many people who have become enthusiastic about this technology are completely uninterested in such pursuits. They want a simple, straightforward, step-by-step guide that'll get them up and running, followed by some sensible examples describing the best methods of implementing common functionality (such as voicemail, auto attendants, and the like).

To the members of the expert community, who (correctly) perceive that Asterisk is like a programming language, this approach doesn't make any sense. To them, it's clear that you have to immerse yourself in Asterisk to appreciate its subtleties. Would one ask for a step-by-step guide to programming and expect to learn from it all that a language has to offer?

Clearly, there's no one approach that's right for everyone. Asterisk is a different animal altogether, and it requires a totally different mindset. As you explore the community, though, be aware that there are people with many different skill sets and attitudes here. Some of these folks do not display much patience with new users, but that's often due to their passion for the subject, not because they don't welcome your participation.

The Asterisk Mailing Lists

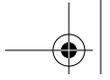
As with any community, there are places where members of the Asterisk community meet to discuss matters of mutual interest. Of the mailing lists you will find at <http://lists.digium.com>, these three are currently the most important:

Asterisk-Biz

Anything commercial with respect to Asterisk belongs in this list. If you're selling something Asterisk-related, sell it here. If you want to buy an Asterisk service or product, post here.

Asterisk-Dev

The Asterisk developers hang out here. The purpose of this list is the discussion of the development of the software that is Asterisk, and its participants



vigorously defend that purpose. Expect a lot of heat if you post anything to this list not relating to programming or development.

Asterisk-Users

This is where most Asterisk users hang out. This list generates several hundred messages per day and has over ten thousand subscribers. While you can go here for help, you are expected to have done some reading on your own before you post a query.

The Asterisk Wiki

The Asterisk Wiki is a source of much enlightenment and confusion. A community-maintained repository of VoIP knowledge, <http://www.voip-info.org> contains a truly inspiring mess of fascinating, informative, and frequently contradictory information about many subjects, just one of which is Asterisk.

Since Asterisk documentation forms by far the bulk of the information on this web site, and it probably contains more Asterisk knowledge than all other sources put together (with the exception of the mailing-list archives), it is commonly referred to as the place to go for Asterisk knowledge.

The IRC Channels

The Asterisk community maintains Internet Relay Chat channels on irc.freenode.net. The two most active are *#Asterisk* and *#Asterisk-Dev*. To cut down on spam-bot intrusions, both of these channels now require registration to join.

The Asterisk Documentation Project

The Asterisk Documentation Project was started by Leif Madsen and Jared Smith. Many people in the community have contributed.

The goal of the documentation project is to provide a structured repository of written work on Asterisk. In contrast with the flexible and ad hoc nature of the Wiki, the Docs project is passionate about building a more focused approach to various Asterisk-related subjects.

As part of the efforts of the Asterisk Docs project to make documentation available online, this book is available at the <http://www.asteriskdocs.org> web site, under a Creative Commons license.

The Business Case

It is very rare to find businesses these days that do not have to reinvent themselves every few years. It is equally rare to find a business that can afford to replace its





communications infrastructure each time it goes in a new direction. Today's businesses need extreme flexibility in all of their technology, including telecom.

In his book *Crossing the Chasm* (HarperBusiness), Geoffrey Moore opines, "The idea that the value of the system will be discovered rather than known at the time of installation implies, in turn, that product flexibility and adaptability, as well as ongoing account service, should be critical components of any buyer's evaluation checklist." What this means, in part, is that the true value of a technology is often not known until it has been deployed.

How compelling, then, to have a system that holds at its very heart the concept of openness and the value of continuous innovation.

This Book

So where to begin? Well, when it comes to Asterisk, there is far more to talk about than we can fit into one book. For now, we're not going to take you down all the roads that the über-geeks follow—we're just going to give you the basics.



In Chapter 1, we cover some of the engineering considerations you should have in mind when designing a telecommunications system. You can skip much of this material if you want to get right to installing, but these are important concepts to understand, should you ever plan on putting an Asterisk system into production.

Chapter 2 covers obtaining, compiling, and installing Asterisk, and Chapter 3 deals with the initial configuration of Asterisk. Here we cover the important configuration files that must exist to define the channels and features available to your system. This will prepare you for Chapter 4, where we introduce the heart of Asterisk, the dialplan. Having covered dialplan basics, Chapter 5 introduces some more advanced dialplan concepts.



We will take a break from Asterisk in Chapter 6, and discuss some of the more important technologies in use in the PSTN. Naturally, following the discussion of legacy telephony, Chapter 7 discusses Voice over IP.

Chapter 8 introduces one of the more amazing components, the Asterisk Gateway Interface (AGI). Using Perl, PHP, and Python, we demonstrate how external programs can be used to add nearly limitless functionality to your PBX. In Chapter 9, we briefly cover what is, in fact, a rich and varied cornucopia of incredible features and functions, all of which are part of the Asterisk phenomenon. To conclude, Chapter 10 looks forward, predicting a future where open source telephony completely transforms an industry desperately in need of a revolution. You'll also find a wealth of reference information in the book's five appendixes.

This book can only lay down the basics, but from this foundation, you will be able to come to an understanding of the concept of Asterisk—and from that, who knows what you will build?