

5

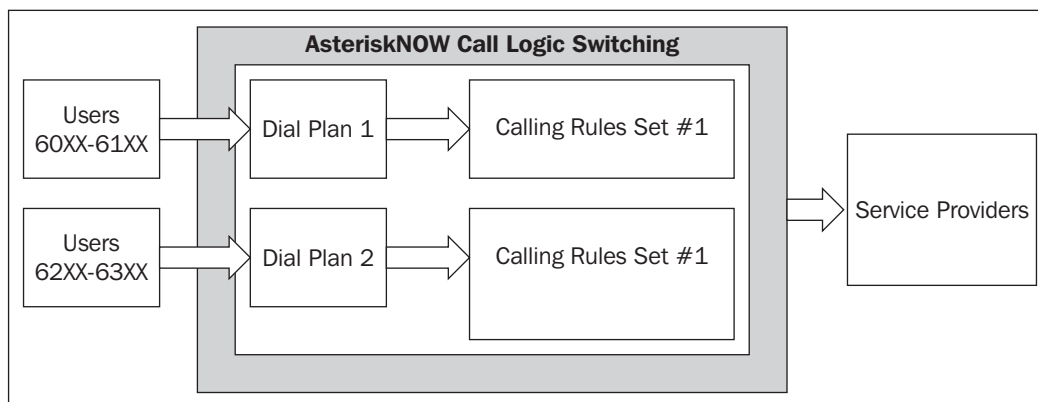
Tentacles of the PBX—The Calling Rules Tables

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.

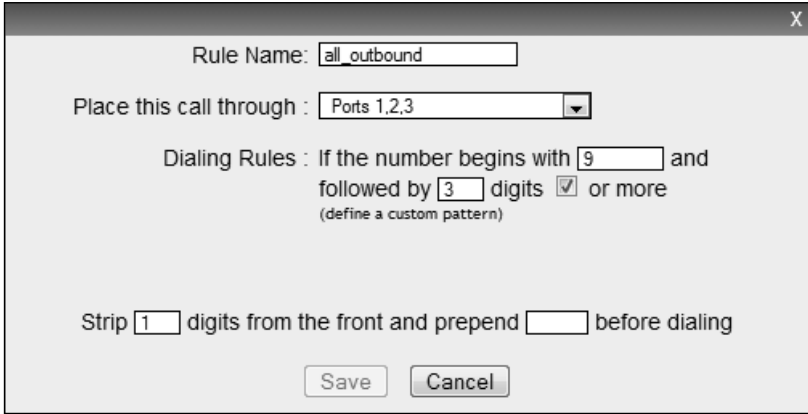
 Most telephony engineers and carrier support engineers refer to "Calling Rules" as "Routing Rules". From this point onwards in the book "Calling Rules" will be referred to as "Routing Rules".

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	AsteriskNOW_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 dialed 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 dialed 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!

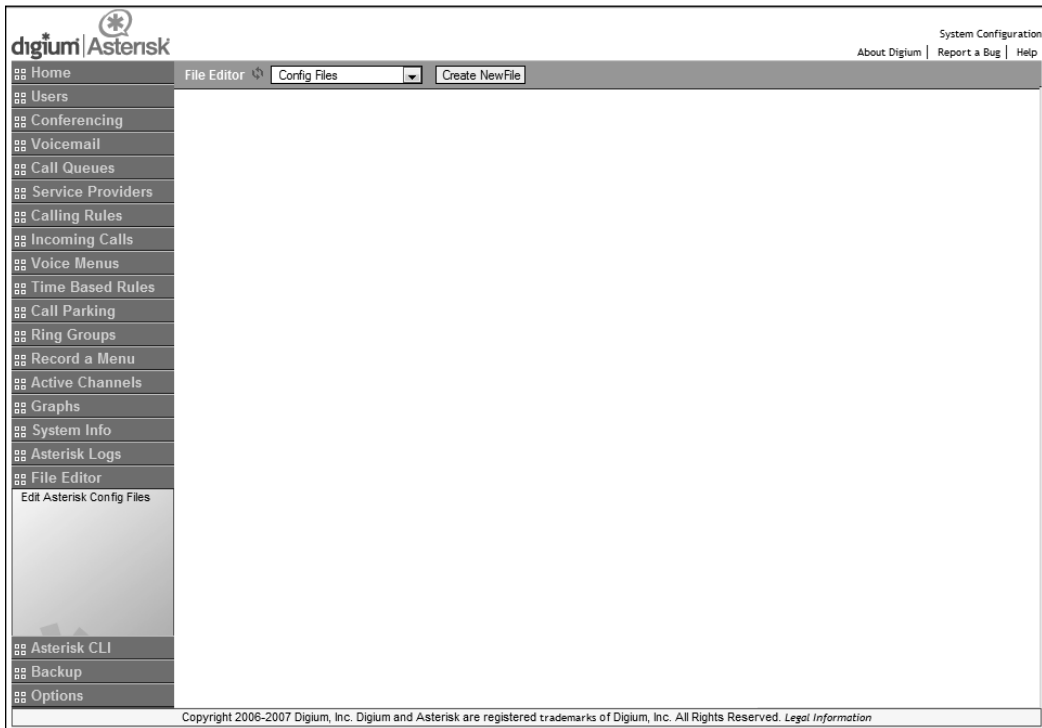
The following section is for advanced users only! If you are not familiar with programming languages, at this point, please skip to the next chapter.



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:

```

extensions.conf Add Context

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

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

[dundi-e164-canonical]

[dundi-e164-customers]

[dundi-e164-via-pstn]

[dundi-e164-local]
include=dundi-e164-canonical
include=dundi-e164-customers

```

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.

```

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

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


```

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.