

Cisco IOS Voice Translations – Part 3: Building and Applying Advanced Expressions in Voice Translation Profiles

By Maren Mahoney, Sunset Learning Cisco Unified Communications Specialized Instructor

This is the third part of a series of four articles on Cisco IOS Voice Translations. In the first part, I covered how to read and write simple regular expressions to construct individual translation rules. The second part covered how rule sets and voice translation profiles are built and applied. This post will discuss advanced regular expressions and special applications of voice translation rules. In the final post in this series, we will look at the tools available to test and troubleshoot voice translations.

Here are the links to the other posts that have been written so far:

- [Cisco IOS Voice Translations – Part 1: The Basics of Voice Translation Rules](#)
- [Cisco IOS Voice Translations – Part 2: Writing and Applying Voice Translation Rules and Voice Translation Profiles](#)

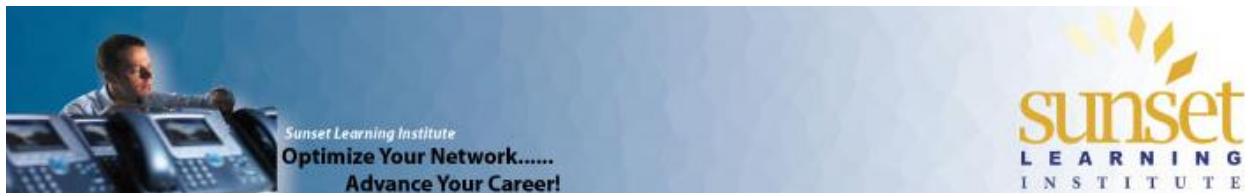
Using a Reject Rule

A “reject” rule in a voice translation ruleset will reject a call matching the parameter(s) listed. The parameter could be something specific, like blocking an inbound call based on a specific caller-ID (shown later) or could block all calls by using a match string of simply // .

First we will look at how to use a generic match string to add a “reject all others” rule to a ruleset. What would happen if a call came in over our circuit where the dialed number is not one of our DID numbers? For instance, hackers can use calls with zero dialed digits (the called-party field is blank) to commit toll-fraud.¹ In a case like this, we may want to add a rule to the ruleset we created in Part 2 (the one that reduces DID numbers to extensions) that will reject any call that does not match one of the other rules (is not for one of our DIDs).

¹ See the following Cisco document for additional information on toll-fraud prevention: "Toll-Fraud Prevention Feature in IOS Release 15.1(2)T"
http://www.cisco.com/en/US/tech/tk652/tk90/technologies_tech_note09186a0080b3e123.shtml





voice translation rule 1

```
rule 1 /7035557.../ /7.../
rule 2 /9725553[0-4].. /3.../
rule 3 /4065559[23].. /5.../
rule 4 reject //
```

Next, what about blocking inbound calls based on caller-ID? This is surprisingly simple and you already know how to do it! Below is our “prefix access codes to incoming caller-ID” voice translation ruleset that we created in Part 2. I have added two rules to the ruleset to reject calls with a specific caller-ID:

voice translation-rule 2

```
rule 1 reject /9005551212/
rule 2 reject /7039761234/
rule 3 /^.....$/ /9&/
rule 4 /^[2-9]..[2-9].....$/ /91&/
rule 5 /^1[2-9]..[2-9].....$/ /9&/
rule 6 /.*/ /9011&/
```

The above rule was applied inbound on our ISDN voice-port to calling-party. With the new ruleset, our circuit will reject calls from those two caller-IDs, even if the dialed number matches one of our DID numbers in voice translation-rule 1 shown above. However, if the caller-ID did not match the numbers from rule 1 and rule 2, the other rules would still be applied.

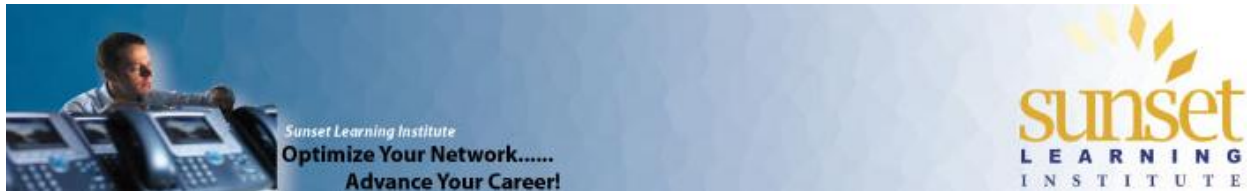
Identifying Calls Based on Type of Number (TON):

The voice translation-rule 2 shown above is one way to prefix access codes to caller-ID on inbound calls. However, it is not considered best practice because it is too specific. For example, malformed caller-ID (think: telemarketer) would be seen as an international number because it does not match another rule. A better way to properly identify, and optionally prefix, caller-ID is to look at the Type of Number (TON) field.

First, a word about what TON means:

Here is the caller ID of an incoming call. Where did the call originate?: 2029871234





Area code 202 is Washington D.C., but what if I told you that Egypt has a country code of 20? Without additional information you cannot know whether this is a US call or an international call.² When sending calls to, or receiving calls from, a service provider an additional piece of information is sent along with the digit string indicating what type of number it is. Types of Numbers (TONs) are:

- International (leading digits are a country code)
- National (internal to the country of the call recipient, but possibly long-distance)
- Subscriber (internal to the area code or region code of the call recipient)
- Unknown

So, for our 2029871234 number, if the TON was “International” we would know that the call originated in Egypt. With a TON of “National” we know the call originated in Washington D.C.

When working with TON in voice translations, the TON can be identified and action taken based on the setting. The TON can also be set or modified for both inbound and outbound calls. The syntax for match/replace of TON is similar to that of the digits themselves:

```
rule 1 /match-digits/ /replace-digits/ type /match-TON/ /replace-TON/
```

In order for the above rule to be executed, both the match-digits string *and* the match-TON would have to match. If a call matches on both parameters then both the replace-digits and the replace-TON would be executed. So, for our voice translation that prefixes access codes to caller-ID, we can simply tell the router to match on TON – regardless of the number itself:

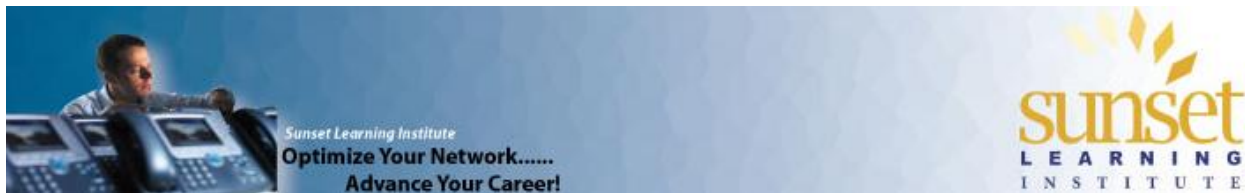
```
voice translation-rule 2
```

```
rule 1 /.*/ /9&/ type subscriber subscriber
rule 2 /.*/ /91&/ type national national
rule 3 /.*/ /9011&/ type international international
rule 4 /.*/ /9&/ type unknown unknown
```

An interesting quirk of primary-ni ISDN circuits is that they will set the TON for a digit-string that begins with 011 to TON=International. However, at the same time they do not remove the 011. This means that the service provider will receive both the 011 prefix (indicating, per the NANP dialplan an international

² In printed materials and in most call-processing systems (such as Cisco Unified Communications Manager), a digit-string generally does not have a TON field associated with it. In these cases, E.164 formatting is often used. E.164 format is also called “plus” format because it uses a plus sign “+” in front of the digit string. The plus sign indicates that the leading digits are a country code. So, our Washington D.C. number would look like this: +12029871234 (the United States is country-code 1) and the call from Egypt would look like this: +209871234.





number) and also the TON international. Some service providers will accept this duplication of information, others will not. To force the primary-ni ISDN circuit to mark the TON=Unknown when the digit string begins with 011, the translation rule might look like this:

```
voice translation-rule 3
  rule 1 /^011.*/ /&/ type international unknown
```

```
voice translation-profile fix-011-out
  translate called 3
```

```
voice-port 0/0/0:23
  translation-profile outgoing fix-011-out
```

using “callback number” as part of translation profile to have callback number display and storage on phone to be different (e.164 format vs. useful format)

Translating Redirect-Called information:

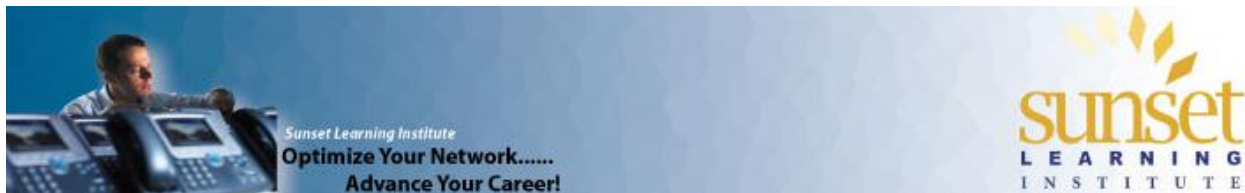
So far we have been discussing called-party and calling-party digit manipulation in voice translations. Another piece of information that may be included in call information is redirect-called information, also known as RDNIS. RDNIS is included in call information when a call has forwarded itself, based on programming, to another number. A common example is forwarding your phone to voicemail on busy or no answer conditions. When the call is forwarded to voicemail, the “new” dialed number (DNIS) for the call is the voicemail system itself. In order for the voicemail system to know what mailbox to send the call to, the original called-party (RDNIS) and often also the original calling-party (R-ANI) are forwarded to the voicemail system along with the reason the call was forwarded (busy, no answer, or unreachable).

For modern telephony systems, the number the caller originally dialed may not match the number associated with the user’s voicemail box. For instance, the redirect number may come in as a full 10-digit DID number, but the user’s voicemail box is configured as a site-code plus extension. (ex.: 703-555-1234 vs. 812-1234) In this case, the dial-peer sending calls to the voicemail system would have to modify the redirect-called information so that the correct voicemail box is accessed.

You already know how to modify numbers using voice translation-rules:

```
voice translation-rule 4
  rule 1 /^703555/ /812/
```





To place the voice translation-rule into a voice translation-profile to modify the redirect-called information, it is as simple as telling it to do so:

```
voice translation-profile fix-voicemail  
  translate redirect-called 4
```

Finally, apply the voice translation profile outbound on the dial-peer sending calls to voicemail.

The three posts in this series so far have given you all of the building blocks to create (and read!) voice translations in Cisco IOS. Applications of voice translation to accomplish certain goals (such as blocking calls based on caller-ID) are all just variations in the application of these rules. If you can figure out how to match, you can modify!

The final post in this series will discuss tools to test voice translation rules and rulesets and examine some of the debugs used to troubleshoot voice translations.

