Sunset Learning Institute
Optimize Your Network... Advance Your Career!
sunset LEARNING INSTITUTE

Cisco IOS Voice Translations – Part 1: The Basics of Voice Translation Rules

Maren Mahoney, Sunset Learning Institute Technical Instructor

Many of my students find that reading (let alone writing) voice translation rules and profiles in Cisco IOS very confusing. And with good reason – the regular expressions used to build the rules themselves take some getting used to. Add to that the application of a set of those rules to a port, dial-peer, etc. – in a direction of a call's flow, both inbound and outbound – for called number, calling party and redirect numbers – and that you can use rule sets to manipulate each of those numbers multiple times for a single call, and the whole voice translation process can become overwhelming. (Aspirin anyone?)

This is the first of a series of articles on Cisco IOS Voice Translations. In this part, I'll cover how to read and write simple regular expressions to construct individual translation rules. Future posts will cover how rule sets and voice translation profiles are built and applied, advanced regular expressions and special applications of voice translation rules, and tools you can use to test what you have built.

The building blocks of voice translations in IOS are:

1) Rule: A single match/replace rule. (This is the focus of this post.)
2) Voice Translation Rule: A set of up to 15 individual rules. A Voice Translation Rule will be used to match numbers in a called party, calling party or redirect number. Since it is a set of rules, it can be used to match one or more patterns of numbers and have each pattern manipulated in a different way.
3) Voice Translation Profile: A set of one or more Voice Translation Rules. A Voice Translation Profile allows you to indicate one Voice Translation Rule for called party, one Voice Translation Rule for calling party and one Voice Translation Rule for redirect number. A Voice Translation Profile can have any or all of these Voice Translation Rules, but you can have only one Voice Translation Rule for each type of number in a Voice Translation Profile.
4) Application of a Voice Translation Profile: Profiles are most commonly applied to ports and dial-peers. When a profile is applied, a direction is declared. One profile can be applied to calls inbound to the port or dial-peer, and another can be applied to calls that are processed through that same port or dial-peer when flowing outbound.

If that is confusing, keep in mind that we will be taking all of that one step at a time. Let's start by looking at how rules are created. First, it's important to understand that a rule is a match/replace instruction. So an individual rule can be read as "if you see this" then "replace it with this". The two expressions are bracketed by the "/" symbol:

```
rule 1     /match-this/       /replace-with-this/
```

Sunset Learning Institute
Optimize Your Network... Advance Your Career!
sunset
LEARNING
INSTITUTE

So let's suppose that you need to strip digits from a 10-digit DID (direct-inward-dial) number and turn it into a 4-digit extension. For example, if the PSTN sends 703-555-7213 to your router and the extension that should ring is 7213 we need to create a rule that says "if you see 7035557213" then "replace it with 7213". Here is what the rule would look like:

```
rule 1     /7035557213/     /7213/
```

To make the rule more functional, we can use wildcards to match all of the 7XXX extensions. A period "." can be used to indicate a single digit. Now the rule looks like this:

```
rule 1     /7035557…/     /7…/
```

Since rules are match/replace, we are really matching only the first seven digits. So another way to write the rule would be:

```
rule 1     /7035557/     /7/
```

Notice that since the match-string includes the 7 it will get stripped out of the number and *replaced* by the replace-string. So the replace-string has to add the 7 back in or you would end up with just the "213".

Another thing to notice in that example is we don't indicate what to do with the "213" at the end of the number. The way translation rules work is that any number not matched in the match-string is automatically included in the replace-string in the same position.

A similar rule can be used to take the 4-digit extension and turn it into a 10-digit caller-id for an outbound call. The rule would read, "if you see 7…" then "replace it with 7035557…":

```
rule 1     /7…/     /7035557…/
```

But there is a problem with that rule. What would happen if the phone initiating the call sent fully formed caller-id like 7035557213 to the router? In that case, the rule would match that string **twice** and would therefore perform the replace twice. The first four digits "7035" would match the match-string and the rule would replace it with the replace-string, and then the last four digits "7213" would also match the match-string and the rule would replace those four digits with the replace string as well. So, **7**03555**7**213 would end up like this:

7035   55   **7**213 ==> **7035557**035   55   **7035557**213 or 70355570355570355557213

Not proper caller-id.

So it would be nice if we could specify that we want to match exactly four digits beginning with 7. This can be done a pair of arguments. A carat symbol "^" means "begins with" and a dollar sign "$" means "ends with". This means that we could re-write our caller-id rule like this:

```
rule 1     /^7…$/     /7035557…/
```

Sunset Learning Institute
Optimize Your Network... Advance Your Career!
sunset
LEARNING
INSTITUTE

It would match a caller-id of exactly four digits, beginning with 7 and turn it into proper caller-id. If the incoming call already had 10-digit caller-id it wouldn't match the match-string and therefore would not be modified.

Next, let's look at another type of wildcard. What if the extension numbers were 7000-7499 and we needed to write rules to strip the DID down to the extension for incoming calls? We can use brackets "[ ]" to indicate either a list, or a range of numbers. So our inbound DID-to-extension rule could be written like this:

```
rule 1    /7035557[0-4]../    /7…/
```

or like:

```
rule 1    /7035557[01234]../    /7…/
```

Easy, right? Now for something a little more complicated.

There are occasions where you will need to change numbers in the middle of the string. One example might be if an organization is employing tail-end hop-off to a location which uses 7-digit dialing. For instance, if a caller in Dallas calls a PSTN number in Montana (which has a single area code and uses 7-digit dialing to local PSTN numbers) and the organization sends the call VoIP to the router in Montana for routing to the PSTN. The user might dial 9-1-406-555-1234, but the digits sent to the router in Montana should be 9-555-1234.

This means that we need to isolate the 9, then the 1-406, then the remaining seven digits so that we can extract the 1-406. This is done with the use of parentheses. To begin, we are going to look at the match-string only:

```
/  (9)   (1406)   (…….)  /
```

However, IOS needs to be told that the open- and close-parentheses are not dialable characters. This is done with an escape character "\". Both the open- and close-parentheses need to be preceded by the escape character. So our match string now looks like this:

```
/  \(9\)   \(1406\)   \(…….\)   /
```

Or, if we put it all together:

```
/\(9\)\(1406\)\(…….\)/
```

The \(9\) is considered "Group 1", the \(1406\) is considered "Group 2" and the \(…….\) is considered "Group 3". The group numbers come into play when we are defining the replace-string. The replace string should read "use Group 1 and then use Group 3". So, it could look like this:

```
/13/
```

except that IOS would see that as thirteen rather than Group 1 and Group 3. To indicate that the numbers 1 and 3 have special meaning, we use the escape character again:

```
/\1\3/
```

Sunset Learning Institute
Optimize Your Network... Advance Your Career!
sunset LEARNING INSTITUTE

So our rule would end up looking like this:

```
rule 1   /\(9\)\(1406\)\(…….\)/   /\1\3/
```

And a dialed number of 9-1-406-5551234 would be translated into 9-5551234. That's all there is to it.

In the next post we will look at how to put rules together into Voice Translation Rules and Voice Translation Profiles, and how to apply Voice Translation Profiles to ports, dial-peers etc.