Voice Translation Rules in Cisco IOS Gateways

To be successful in configuring a voice translation rule in a Cisco IOS gateway, you need to be familiar with three steps. The first step is to configure a voice translation rule. The second step is to configure a voice translation profile and to associate a voice translation rule with the voice translation profile. The third step is to apply the voice translation profile to a voice port or a dial peer.

Step 1: Configure the Individual Rules

The first command you will use is the **voice translation-rule** *number* command in global configuration mode, which will match and replace digits. Use this command to define the rule that will be associated with a voice translation profile. The number is a unique identifier that is used to associate a set of translation rules with a translation profile. Here is an example of the use of the **voice translation-rule** command.

```
Router(config)#voice translation-rule 1
```

Refer to Table 1 for the options associated with the **voice translation-rule** command.

Table 1

Syntax	Description
number	Number that identifies the translation rule. Range is from 1 to 2147483647.

Several rules can be grouped together into a single translation rule. The precedence portion of the command sets the priority of the translation rule. The precedence value of each rule can be used in a different order from that in which the rules were typed into the set. The precedence value of each rule specifies the priority order in which the rules are to be used. For example, rule 3 may be entered before rule 1, but the software uses rule 1 before rule 3.

A translation rule can contain up to 15 rules. All calls that refer to this translation rule are translated against this set of rules. A good way to check how the translation rules are matched is to write a voice translation rule with multiple rules, each with a different match and replacement string. What you will find is that if the dialed string matches a voice translation rule with multiple rules, the string will search the rules until it finds the first match and stops. Then use the **test voice translation-rule** *number* command and a dialed string. The output from the test command and the debug will show you which of the rules were matched and will provide details on what was matched and what was replaced. If the dialed string fails to match any of the rules in a voice translation rule, a message saying “Didn’t match with any of rules” will be returned. There are no default values when you configure the rule.

Here is an example of the use of the **rule** command and some of the associated options:

```
Router(cfg-translation-rule)#rule 1 /1234/ /5678/ plan national national type
unknown isdn
```

Refer to Table 2 for the options associated with the **rule** command.

Table 2

Syntax	Description
precedence	This value sets the priority of the translation rule. The range is from 1 to 15.
| *match–pattern* |	This is the format of a matching pattern. The "/" is used to mark the start and end of the character string you want to match.
| *replace-pattern* |	Notice that this is really the same command syntax, but here you will create a string of characters that will replace what was matched in the matching pattern. *Wildcards are not permitted in the replacement string, because they are in the match string.*
type match-type replace-type	This is an optional subset of the rule command that allows the numbering match to be as follows: • **abbreviated**—Abbreviated representation of the complete number as supported by this network. • **any**—Any type of called number. • **international**—Number called to reach a subscriber in another country. • **national**—Number called to reach a subscriber in the same country, but outside the local network. • **network**—Administrative or service number specific to the serving network. • **reserved**—Reserved for extension. • **subscriber**—Number called to reach a subscriber in the same local network. • **unknown**—Number of a type that is unknown by the network. Valid values for the *replace-type* argument are the same as in the preceding list except for "any," which is not valid for this type of argument.

plan *match-type replace-type*	(Optional) Numbering plan of the call. Valid values for the *match-type* argument are as follows: • **any**—Any type of dialed number. • **data** • **ermes** • **isdn** • **national**—Number called to reach a subscriber in the same country, but outside the local network. • **private** • **reserved**—Reserved for extension. • **telex** • **unknown**—Number of a type that is unknown by the network. Valid values for the *replace-type* argument are the same as in the preceding list except for "any," which is not valid for this type of argument.

Match and replacement strings use regular expressions to match and replace digits. In the preceding example, rule 1 has simple match and replacement strings. These strings can become quite complex, and the use of regular expressions can be confusing.

There are some foundational rules that can help you understand how to configure a voice translation rule using the voice translation rule operators:

- You begin and end any match or replacement string with the "/" operator.
- A match string can be made up of explicitly matched digits and wildcards. Wildcards are allowed only in the match string.
- If you want a match string to match from the beginning of a string of digits, the caret (^) is the first operator in the match string.
- If you want to match from the end of a match string, the dollar sign ($) is the first character.

- The slash (\) rule operator delimits the next character in the string. For example, in the match string /^3\(…\)/, the first "\" rule operator tells the router to not match or replace the next character, which is the "(". The second "\" also tells the router to not match or replace the next character, which is the ")".

- If you want to match a group of wildcards or explicit digits and not change them, you can create a set. For example, the match string /^3\(…\)/ matches any dialed string that begins with 3. The (…\) creates the first set where any of the next three dialable digits are matched. In another example, the replacement string /4\1/ will replace any explicitly matched digit with a 4, and the "\" will cause the router to view the 1 as a control character and will cause the contents of the first set to be pulled across to the replacement string. The complete rule would look like this: rule 1 /^3\(…\)/ /4\1/

- Any digits that are part of a match string and are not explicitly matched by the match string are carried across to the replacement string unchanged.

Table 3 lists and describes the individual operators that you can use in a voice translation rule.

Table 3

Voice Translation Rule Operators	Description
^	Matches the expression at the start of the line.
$	Matches the expression at the end of the line.
/	Delimiter that marks the start and end of both the matching and replacement strings.
\	Escapes the meaning of the next character.
-	Indicates a range when not in the first or last position. This is typically used with the [and] to indicate a range.
[list]	Matches a single character specified in a list.
[^list]	Does not match a single character specified in a list.
.	Matches any single character.
*	Repeats the previous regular expression zero or more times.
+	Repeats the previous regular expression one or more times.
?	Repeats the previous regular expression zero or one occurrence of the previous expression. (Use **Ctrl-V ?** to enter in IOS.)
()	Groups digits into sets.
&	Brings all the matched digits into the replacement string.

Examples of Regular Expression Usage in Voice Translation Rules

Example 1: Matching the NULL String

```
Router(config)#voice translation-rule 1
Router(cfg-translation-rule)#rule 1 /^$/ / /
```

Match String	Replace String	Dialed String	Replaced String
/^$/	/ /	NULL	NULL

You first need to understand what the NULL string is before you can understand what this rule is doing. A NULL string is an empty set. It is not a string that matches nothing; it is a string that matches an empty string of digits. For example, you may not receive caller ID from the public switched telephone network (PSTN). The NULL match string would match the empty caller ID. In this example the match string is a NULL string and the replacement string will carry across whatever is matched in the match string. The match string starts with the "/". Since you want to create a NULL match string, the next operator "^"begins matching from the beginning of the string. The next character is the "$," which begins matching from the end of the string. This is followed by the "/", which ends the match string. This match string will be empty.

Now look at the replacement string. It starts with the "/" and is followed by a space and then another "/". There are no characters in the replacement string, so it will replace with the contents of the match string, which is empty.

If you test this voice translation rule, the output is as follows:

```
Router#test voice translation-rule 1 7701234
7701234 Didn't match with any of rules
```

Example 2: Matching the NULL String and Replacing with Other Digits

```
Router(config)#voice translation-rule 1
Router(cfg-translation-rule)#rule 1 /^$/ /2001/
```

Match String	Replace String	Dialed String	Replaced String
/^$/	/ 2001/	NULL	2001

In this example, you are continuing to match the NULL string. You may have a situation where you are not receiving caller ID from the PSTN or you may have analog lines that do not provide DNIS. You would prefer having these calls handled by an attendant rather than matching the default dial peer. In the preceding example, you match on the NULL string, which means you match on the empty digit field that the PSTN may be sending you for caller ID or an empty DNIS field on an analog circuit. You then replace the empty match string with the digits for the internal attendant. In this example, the internal attendant is located at extension 2001. After the replacement is made, a recursive lookup of the new dialed string is made of the internal dial peers to attempt to match the extension with 2001. If there is a match, the call is sent to the appropriate extension.

Example 3: Replacing a Match String with the Any String

```
Router(config)#voice translation-rule 1
Router(cfg-translation-rule)#rule 1 /^.$/ / /
```

Match String	Replace String	Dialed String	Replaced String
/^ . $/	/ /	9	nothing

The match string is slightly different; it now includes the "." (period). The "."causes any single character to be matched. Since the first character in the match string is the "^," you match the 9 in the dialed string. The replacement string is again "/" followed by a space and then another "/". This causes the replacement string to replace any digits that were not explicitly matched by the match string. Since you have dialed 9 and that was matched, there are no other digits to replace.

If you test this voice translation rule, the output is as follows:

```
Router#test voice translation-rule 1 9
Matched with rule 1
Original number: 9              Translated number:
Original number type: none Translated number type: none
Original number plan: none Translated number plan: none
```

Example 4: Implementing Simple Match and Replacement Strings

```
Router(config)#voice translation-rule 1
Router(cfg-translation-rule)#rule 1 // //
```

Match String	Replace String	Dialed String	Replaced String
//	//	7705550100	7705550100

The match string is empty and so is the replacement string. This is different from a NULL match or replacement string. Here the match string will match anything. Since no digits are explicitly matched, any digits received will flow across to the replacement string. When you dial 7705550100, the fact that you have a string of digits triggers the rule. None of the digits is explicitly matched, so the digits are carried through unchanged.

If you test this voice translation rule, the output is as follows:

```
Router#test voice translation-rule 1 7705550100
Matched with rule 1
Original number: 7705550100     Translated number: 7705550100
Original number type: none      Translated number type: none
Original number plan: none      Translated number plan: none
```


The preceding example is not something that you would typically do, but with a slight modification, this example becomes very useful. Building a rule where the match string is "//" is a simple way to prefix numbers.

```
Router(config)#voice translation-rule 1
Router(cfg-translation-rule)#rule 1 // /9/
```

Match String	Replace String	Dialed String	Replaced String
//	/9/	7705550100	97705550100

This new rule will prefix 9 to any digit string.

Example 5: Using One Set for Match and Replacement Strings

As stated earlier, any digits that are matched are replaced by the contents of the replace pattern. Sometimes it is desirable to match digits that you do not want to change. You can do this by grouping digits or wildcards into sets.

```
Router(config)#voice translation-rule 1
Router(cfg-translation-rule)#rule 1 /^555\(.*\)/ /678\1/
```

Match String	Replace String	Dialed String	Replaced String
/^555\ (.*\)/	/678\1/	5550100	6780100

This is a replacement rule where you want to match a string that begins with 555 and replace the 555 with 678. In this example you use the "^" to match from the beginning of the line. You follow that with 555, so if the dialed string was 4040100, it would not match this rule. As stated earlier, the "\" character means that the next character is a control character and should not be matched or replaced. In the example the next character is the "(". If the "\" rule character was not in front of the "(",the gateway would try to match this character, which it could not do. The rule characters "(" and ")" mark the beginning and end of a set. A set can be made up of explicit characters or wildcards. In the example the set uses the "." wildcard, which will match any single character, and the "*" to repeat the previous regular expression zero or more times. In the example the set will cover the digits received after the 555, which will be 0100 in the dialed string.

Now look at the replacement string. You see that the "/" starts the replacement string. The number 678 is followed by the "\1". The "\" indicates that the 1 is a control character, not a replacement character. The "\1" refers to the first set in the match expression identified as (.*\) and will bring the contents of this set to the replaced string.

If you test this voice translation rule, the output is as follows:

```
Router#test voice translation-rule 1 5550100
Matched with rule 1
Original number: 5550100         Translated number: 6780100
Original number type: none       Translated number type: none
Original number plan: none       Translated number plan: none
```


Example 6: Using Two Sets for Match and Replacement Strings

```
Router(config)#voice translation-rule 1
Router(cfg-translation-rule)#rule 1 /^\(770\)\(....\)/ /678\2/
```

Match String	Replace String	Dialed String	Replaced String
/^\(770\)\(….\)/	/678\2/	7701234	6781234

This example is a variation of the previous rule. Can you spot the differences? In the match string, you start with the "/" to start the match string, and then use the "^" to match from the beginning of the string. Next, you use the "\" to delimit the "(", which begins the first set. Then you want to match 770 followed by the "\" to delimit the ")", which completes the first set in the match string. There is another "\" to delimit the "(", which starts the second set. The second set is made up of four ".". This means you are going to match any of the next four digits. The "\" follows the four dots, and the set is closed by the ")".Which of the following dialed strings would match this rule?

7701234
7709876
6781234
7701234567

The correct answers are all of them except the dialed string that begins with 678. You are probably wondering about the output from a dialed string of 7701234567. The matching string will match the 770 in the first set and the next four digits in the second set, and pull across the last three digits unchanged. As stated earlier, any digits that are not explicitly matched in the match string are pulled across to the replacement string unchanged. Since the digits 567 are not explicitly matched in the match string, they are pulled across unchanged into the replacement string.

If you test this voice translation rule, the output is as follows:

```
Router#test voice translation-rule 1 7701234
Matched with rule 1
Original number: 7701234         Translated number: 6781234
Original number type: none       Translated number type: none
Original number plan: none       Translated number plan: none

Router#test voice translation-rule 1 7709876
Matched with rule 1
Original number: 7709876         Translated number: 6789876
Original number type: none       Translated number type: none
Original number plan: none       Translated number plan: none

Router#test voice translation-rule 1 6781234
6781234 Didn't match with any of rules

Router#test voice translation-rule 1 7701234567
Matched with rule 1
Original number: 7701234567      Translated number: 6781234567
Original number type: none       Translated number type: none
Original number plan: none       Translated number plan: none
```


Example 7: Using One Set for Match and Replacement Strings

```
Router(config)#voice translation-rule 1
Router(cfg-translation-rule)#rule 1 /^770\(....\)/ /678\1/
```

Match String	Replace String	Dialed String	Replaced String
/^770\(….\)/	/678\1/	7701234	6781234

By now you should be seeing a pattern in the match and replacement strings. This is a little easier than the last one. What do you know? Where will you start in the dialed string? What are you going to match? How many sets are there in the matched string? What is the replacement string going to do? Have you got it?

You will start at the beginning of the string because the match string begins with a "^". Next, you are going to match 770 in the string. There is one set identified by "\(….\)". The replacement string will replace 770 with 678, and the digits in set one will be pulled into the replacement string.

If you test this voice translation rule, the output is as follows:

```
Router#test voice translation-rule 1 7701234
Matched with rule 1
Original number: 7701234         Translated number: 6781234
Original number type: none       Translated number type: none
Original number plan: none       Translated number plan: none
```

Example 8: Replacing the Middle of a Match String

```
Router(config)#voice translation-rule 1
Router(cfg-translation-rule)#rule 1 /\(^...\)555\(....\) /\1678\2/
```

Match String	Replace String	Dialed String	Replaced String
/\(^…\)555\(….\)/	/\1678\2/	4085550100	4086780100

This example is a little more difficult to understand. You start with a match string that starts with a set that starts with a "^" and matches the first three numbers. Then you match explicitly 555, followed by a second set that matches the next four digits in the string. Then you finish the match string with the "/".

Now look at the replacement string. This probably looks a little strange at first glance. You start the replacement string with "/" and follow that with "\1". You put the backslash in front of the 1, so that the first set is pulled through from the match string. The 678 will replace the 555 in the match string. Then you delimit the number 2 with "\2" so that the second set is pulled through into the replacement string.

So when the dialed string is something like 4045550100, the replacement is 4046780100. All you did was change the middle three digits of the dialed string.

There are other ways to do this with three sets, where the matched 555 becomes part of a set and the replacement string would then need to change to /\1678\3/, so that the first set is pulled into the replacement string, the 555 is replaced by 678, and then the third set is pulled into the replacement string.

If you test this voice translation rule, the output is as follows:

```
Router#test voice translation-rule 1 4085550100
Matched with rule 1
Original number: 4085550100      Translated number: 4086780100
Original number type: none       Translated number type: none
Original number plan: none       Translated number plan: none

Router#test voice translation-rule 1 4045551234
Matched with rule 1
Original number: 4045551234      Translated number: 4046781234
Original number type: none       Translated number type: none
Original number plan: none       Translated number plan: none
```

Example 9: Matching a Set at the Beginning of a String and Replacing the End of the String

```
Router(config)#voice translation-rule 1
Router(cfg-translation-rule)#rule 1 /\(^.*\)0100/ /\10199/
```

Match String	Replace String	Dialed String	Replaced String
/\(^.*\)0100/	/\10199/	5550100	5550199

This rule matches from the end of the string. Notice that the match string has a single set and that it matches the digits 0100. The replacement string changes the matched digits from 0100 to 1099 and pulls across set 1 from the matched string. If you dial the following strings, which ones will be matched and will use the replacement string?

9195550100
5550100
6780100
7770100
4045550100

The answer is that all these strings will match the match string, and the replacement string will change them to the following:

9195550199
5550199
6780199
7770199
4045550199

The reason is that the first set is \(.*\). This means that you will match any character and then repeat it zero or more times. So if you have a variable-length number that is being dialed but ends in 0100, the replacement string will pull across everything in the first set and change only the last four digits as long as they are 0100.

If you test this voice translation rule, the output is as follows:

```
Router#test voice translation-rule 1 9195550100
Matched with rule 1
Original number: 9195550100      Translated number: 9195550199
Original number type: none       Translated number type: none
Original number plan: none       Translated number plan: none

Router#test voice translation-rule 1 5550100
Matched with rule 1
Original number: 5550100         Translated number: 55501099
Original number type: none       Translated number type: none
Original number plan: none       Translated number plan: none

Router#test voice translation-rule 1 7700100
Matched with rule 1
Original number: 7700100         Translated number: 7700199
Original number type: none       Translated number type: none
Original number plan: none       Translated number plan: none

Router#test voice translation-rule 1 4046780100
Matched with rule 1
Original number: 4046780100      Translated number: 4046781099
Original number type: none       Translated number type: none
Original number plan: none       Translated number plan: none
```

Example 10: Prefixing Digits and Changing the Number Type

```
Router(config)#voice translation-rule 1
Router(cfg-translation-rule)#rule 1/^2/ /912105552/ type unknown national
```

Match String	Replace String	Dialed String	Replaced String
/^2/	/912015552/	2001	912015552001 type national

In this example, you want to expand the dialed digits so that they can place a national call over the PSTN. The match string will match from the beginning of the dialed string and will match any dialed string that begins with a 2. The replace string will prefix the access code of 9 and then dial a 1 for long distance and add the area code of 201 and the office code of 5552. Notice that the replace string includes the leading 2 from the match string. If you fail to include this in the replacement string, the only digits that will be carried across to the replacement string will be the 001 of the dialed string.

If you test this voice translation rule, the output is as follows:

```
Router#test voice translation-rule 1 2001 type national
Matched with rule 1
Original number: 2001            Translated number: 912015552001
Original number type: unknown    Translated number type: national
Original number plan: none       Translated number plan: none
```


Example 11: Prefixing Digits and Changing the Number Type and Number Plan

```
Router(config)#voice translation-rule 1
Router(cfg-translation-rule)#rule 1/^2/ /912015552 type unknown national plan
unknown isdn
```

Match String	Replace String	Dialed String	Replaced String
/^2/	/912015552/	2001	912015552001 type national plan isdn

This example is similar to Example 10. The changes here are in the number type and the number plan. You have changed from an unknown number type to a national number type and from an unknown number plan to an ISDN number plan.

If you test this voice translation rule, the output is as follows:

```
Router#test voice translation-rule 1 2001 type national
Matched with rule 1
Original number: 2001            Translated number: 912015552001
Original number type: unknown    Translated number type: national
Original number plan: unknown    Translated number plan: isdn
```


Step 2: Configure a Voice Translation Profile

A voice translation profile is made up of two commands. First is the command **voice translation-profile** *name*. Use this command in global configuration mode to define a voice translation profile for voice calls. The *name* portion of the command can be up to 31 characters long and differentiates one voice translation profile from another. There is no default value for this command.

Here is an example of a voice translation profile. The descriptive name PSTNout has been assigned to the voice translation profile so that it is easier to determine what the profile and associated rules do.

```
Router(config)#voice translation-profile PSTNout
```

Refer to Table 4 for the configuration of the **voice translation-profile** command.

Table 4

voice translation-profile *name*	This command creates the voice translation profile. The name differentiates one voice translation profile from another and can be up to 31 characters long.

The second part of the voice translation-profile command uses one of four options. They are **translate called** *translation rule***#**, **translate calling** *translation rule***#**, **translate redirect called** *translation rule***#**, and **translate redirect-target** *translation rule* **#**. The requirements of the translation profile will dictate which of these options you need to use. These commands take the voice translation rule that you want to associate with the voice translation profile and define if the called or calling numbers are translated, the called number is redirected, or if the translation rule is to be associated with transfer-to numbers or call-forwarding final destination numbers.

Here is an example where a **voice translation-profile PSTNout** command has been created and **voice translation-rule 4** will be used to translate called numbers.

```
Router(config)#voice translation-profile PSTNout
Router(cfg-translation-profile)# translate called 4
```

Refer to Table 5 for the configuration of the **voice translation-profile** command.

Table 5

Syntax	Description
translate called *translation rule* **#**	This command under the voice translation profile defines the translation profile rule for the called number.
translate calling *translation rule* **#**	This command under the voice translation profile defines the translation profile rule for the calling number.
translate redirect called *translation rule* **#**	This command under the voice translation profile defines the translation profile rule for the redirect-called number.
translate redirect-target *translation rule* **#**	This command associates the translation rule with transfer-to numbers and call-forwarding final destination numbers.

Step 3: Assign a Voice Translation Profile to a Dial Peer or a Voice Port

To assign a translation profile to a dial peer, use the **translation-profile** command in dial-peer configuration mode. Here is an example of creating a voice translation rule, associating it with a voice translation profile and assigning the voice translation profile with a dial peer.

```
Router(config)#translation-rule 1
Router(cfg-translation-rule)#rule 1 /^2…/ /912012012/
Router(config)#voice-translation-profile PSTNout
Router(cfg-translation-profile)#translate called 1
Router(config)#dial-peer voice 910 pots
Router(config-dial-peer)#destination-pattern 2…
Router(config-dial-peer)#port 1/0:0
Router(config-dial-peer)#translation profile outgoing PSTNout
```

In the example, voice translation rule 1 has been created with rule 1. Next, a voice translation profile has been created with the name PSTNout, and voice translation rule 1 has been associated with it using the command **translation profile outgoing PSTNout**. Finally, the voice translation profile has been assigned to the dial peer with the command **translation profile outgoing PSTNout**.

Refer to Table 6 for the configuration of the translation profile in a dial peer.

Table 6

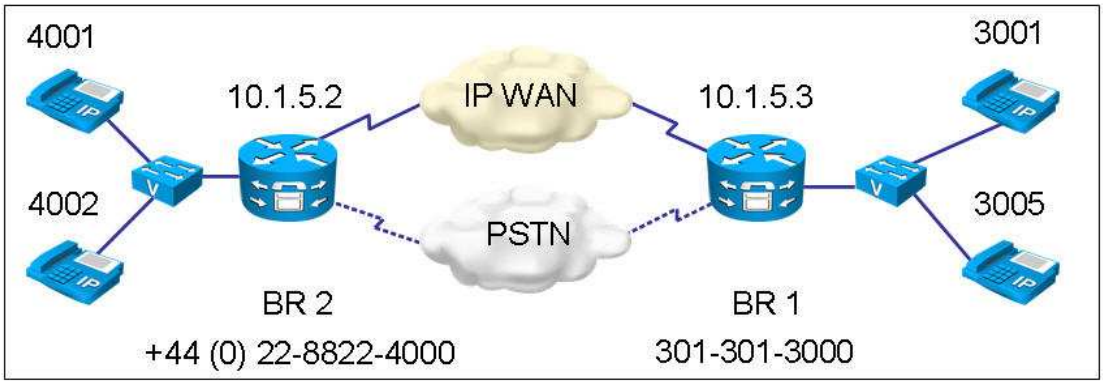
Syntax	Description
incoming	Specifies that this translation profile handles incoming calls.
outgoing	Specifies that this translation profile handles outgoing calls.

To assign a translation profile to a voice port, use the **translation-profile** command in voice port configuration mode. Here is an example of creating a voice translation rule, associating it with a voice translation profile, and assigning the voice translation profile with a voice port.

```
Router(config)#translation-rule  3
Router (cfg-translation-rule)#rule   1 /^.*\(....\)/ /\1/
Router(config)#voice-translation-profile  PSTNin
Router(config)#voice-port 1/0:0
Router (config-voiceport)#translation-profile incoming PSTNin
```

In the example, voice translation rule 3 has been created with rule 1. Next, a voice translation profile has been created with the name PSTNin, and voice translation rule 3 has been associated with it using the command **translate incoming PSTNin**. Finally, the voice translation profile PSTNin has been assigned to voice port 1/0:0 with the command **translation-profile incoming PSTNin**.

Example 1



This is a typical situation where you have the potential for two paths between Branch 2 and Branch 1. In a normal operating situation, the users at Branch 2 use four-digit dialing to reach Branch 1. Please refer to the **dial-peer voice 300 voip** command in the configuration snip that follows. This dial peer will send the calls from Branch 2 to Branch 1 over the IP WAN. What happens when the IP WAN fails? The users at Branch 2 will need to remember the full E.164 telephone number for Branch 1 and the specific extension they need to reach. If the E.164 number matches the extension numbers, this might be simple, but if the organization has 25 remote locations and not every location has an exact match between direct inward dialing (DID) and the extension number, that might cause problems for the end users.

The simple solution is to set up two voice dial peers. Configure one dial peer to send calls over the IP WAN, and configure a second dial peer that has a lower preference to send calls over the PSTN. The problem is that if you want the end users to continue to use four-digit dialing, you need to make provisions in the dial peer that sends calls out to the PSTN to prefix the necessary digits to provide an E.164 number.

Refer to the preceding diagram. The end user at Branch 2 who has extension 4001 wants to place a call to a user at Branch 1 who has extension 3001. The Branch 2 user dials 3001, and the dialed string matches **dial-peer voice 300 voip.** The call is sent over the IP WAN to the device with the address of 10.1.5.3, which is the Cisco Unified Communications Manager Express device at Branch 2. If the IP WAN has failed, the call will need to be routed across the PSTN. This process will require the dialed number to receive a prefix so that it can properly dial the E.164 number associated with the specific extension at Branch 1.

```
!
voice translation-rule 1
rule 1 /^3\(...\)/ /0013013013\1/
!
voice translation-profile branch1
translate called 1
!
dial-peer voice 300 voip
destination-pattern 3...
session target ipv4:10.1.5.3
!
dial-peer voice 3000 pots
destination-pattern 3...
preference 1
translation-profile outgoing branch1
port 1/0:23
!
```


Refer again to the configuration snip. The first line creates the voice translation rule 1 with rule 1. This matches a dialed string that begins with 3 and that contains a set that matches three variable digits. You may also need to consider the position of the digits in the dialed string you are matching.

The replacement string then prefixes the appropriate digits for an international call. In the example, 00 indicates an international call, 1 is the country code for the United States, 301 is the area code, 301 is the prefix, and 3 is the part that was matched in the match string but was not brought across to the match string because it matched the match string explicitly. Finally, the first set is pulled across from the match string to the replacement string to complete the E.164 phone number.

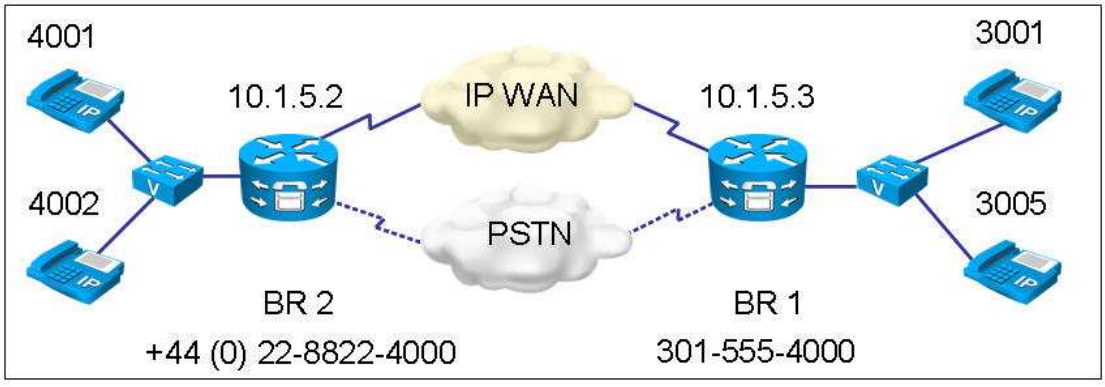
The next component configured is the voice translation profile. Notice that this has been given the name *branch1*, and you will be translating the called number using voice translation rule 1. The **translation-profile outgoing branch1** command has been applied to **dial-peer voice 3000 pots**.

When a user at Branch 2 places a call to Branch 1 and the IP WAN is down, **dial-peer voice 3000 pots** will be matched, since it has the same destination pattern as **dial-peer voice 300 pots** but a lower preference. The **voice translation-profile outgoing branch1** command will translate the dialed digits 3001 to 0013013013001 and send the call out to the PSTN.

There is an alternative method of achieving the same result but without the voice translation rule. You could create two dial peers that point to the IP WAN and the PSTN with the highest preference going to the IP WAN dial peer. You could then use the prefix command in place of the voice translation rule to prefix the required digits to place the call over the PSTN. The following configuration example shows how this is accomplished.

```
!
dial-peer voice 300 voip
destination-pattern 3...
session target ipv4:10.1.5.3
!
dial-peer voice 300 pots
destination-pattern 3...
prefix 9001301301
port 1/0:23
preference 1
!
```

Example 2



In this scenario, Branch 1 has moved. The area code is the same, but the main number for the office has changed, although the extension numbers remain the same. Calls from Branch 2 to Branch 1 can still use 4-digit dialing over the IP WAN, but calls that are placed over the PSTN now fail because the DID range is different from the extension numbers.

Refer to the configuration snip. A **voice translation-rule 1** command has been created with rule 1 that matches any string beginning with a 3 followed by three variable digits. This is translated to an international number where 00 is the international dialing code, 1 is the country code for the United States, the area code is 301, the prefix is 555, and the first digit of the DID range is 4. A voice translation profile has been created with the name branch1pstn, and the profile will translate outgoing dialed digits that begin with a 3 using **voice translation–rule 1**.

The dial peer that places calls over the IP WAN will continue to work. The dial peer to send calls over the PSTN is configured with the **voice translation-profile outgoing branch1pstn** command.

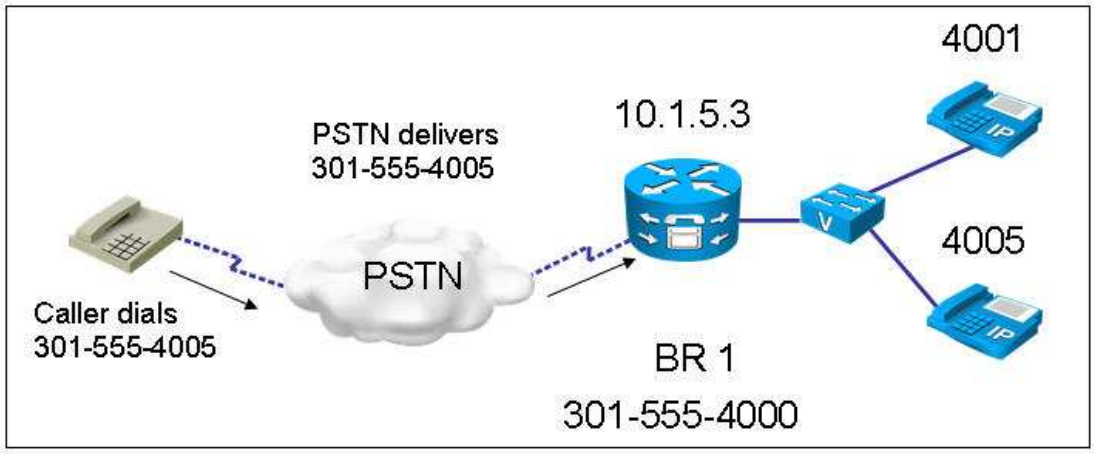
```
!
voice-translation rule 1
rule 1 /^3\(...\)/ /0013015554\1/
!
voice translation-profile branch1pstn
translate called 1
!
dial-peer voice 300 voip
destination-pattern 3...
session target ipv4:10.1.5.3
!
dial-peer voice 3000 pots
destination pattern 3...
preference 1
translation-profile outgoing branch1pstn
port 1/0:23
!
```

At Branch 1 the PSTN is providing the 10-digit string 301-555-4000. Since the internal extensions no longer match the DID range, a voice translation rule and voice translation profile must be created to reduce the 10 digits to 4 digits and to change the leading extension number from the DID range starting with 4000 to 3000.

Refer to the configuration snip. First, a voice translation rule 1 is created with rule 1 that matches the 10-digit DID range and strips it down to 4 digits. At the same time, the rule changes the leading digit from a 4 to a 3. Now if the IP WAN fails, any branch that tries to dial across the PSTN will reach the correct extensions without dialing any extra digits.

```
!
voice translation-rule 1
rule 1 /^3015554\(...\)/ /3\1/
!
voice translation-profile PSTNin
translate incoming 1
!
voice-port 1/0:23
translation profile incoming PSTNin
!
```


Example 3



You will now apply a voice translation rule to a voice port. You may need to do this in situations where the PSTN is delivering 10 digits and you need to strip them down to 4 digits so that they can be matched to the internal extensions. In the example a caller dials 301-555-4005 to place a call to extension 4005 at Branch 1. The PSTN delivers the call to Cisco Unified Communications Manager Express at Branch 1. The voice translation rule that meets these requirements is as follows:

```
!
voice translation-rule 3
rule 1 /^301555\(....\)/ /\1/
!
voice translation-profile incoming pstn
translate called 3
!
```

Now you need to apply the voice translation rule to the voice port. The command syntax to complete this step is as follows:

```
!
voice-port 1/0:0
translate called pstn
!
```

In many gateway configurations, the following dial-peer configuration is included. This dial peer matches any incoming number, and the **direct-inward dial** command keeps the number from having a secondary dial-tone.

```
!
dial-peer voice 1 pots
incoming-called number .
direct-inward dial
!
```


If the PSTN is sending 10 digits, the gateway will not match any of the internal dial peers, but the 10 digits may match either **dial-peer voice 1 pots** if it is configured or the default dial peer. If **dial-peer voice 1 pots** is not configured, the incoming 10 digits will match only the default dial peer and will return a fast busy signal to the caller, because the number dialed does not match any internal dial peers. By using the voice translation profile and applying it to the voice port to strip the 10-digit number being presented by the PSTN to 4 digits, the gateway can now match them against the internal dial peers.

Even if you omit **dial-peer voice 1 pots** from the configuration and configure a voice translation profile to strip the 10-digit number to 4 digits, the number will match an internal dial peer.

